

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
10 May 2001 (10.05.2001)

PCT

(10) International Publication Number  
**WO 01/33430 A1**

(51) International Patent Classification<sup>7</sup>: **G06F 17/30**,  
7/00, 13/00, 3/00

[IL/IL]; Hayetzia Street 3, 52118 Ramat-Gan (IL).  
**BREGMAN, David** [US/US]; 560 Hilmar Street, Santa Clara, CA 95050 (US). **EMBERSON, Richard** [US/US]; 2656 Kipling, Palo Alto, CA 94306 (US). **POBER, Jay** [US/US]; 1128 Fulton Street, Palo Alto, CA 94301 (US).

(21) International Application Number: PCT/US00/29719

(74) Agents: **MALLIE, Michael, J.** et al., Blakely, Sokoloff, Taylor & Zafman LLP, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, CA 90025 (US).

(22) International Filing Date: 27 October 2000 (27.10.2000)

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(25) Filing Language: English

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European

(26) Publication Language: English

*[Continued on next page]*

(30) Priority Data:

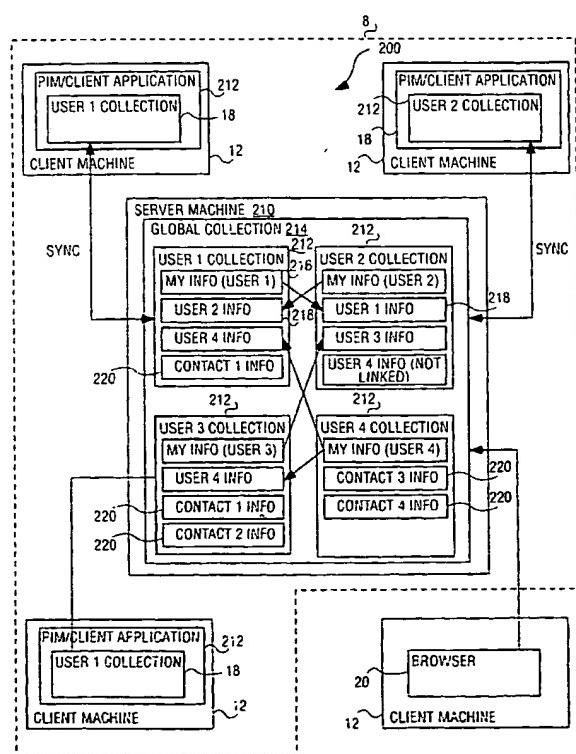
60/162,499	29 October 1999 (29.10.1999)	US
09/566,053	5 May 2000 (05.05.2000)	US
09/680,130	2 October 2000 (02.10.2000)	US

(71) Applicant (*for all designated States except US*): **CONTACT NETWORKS, INC.** [US/US]; 2600 Bridge Parkway, Suite 101, Redwood Shores, CA 94065 (US).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **HERTZOG, Eyal**

(54) Title: METHOD AND SYSTEM FOR UPDATING USER INFORMATION MAINTAINED BY ANOTHER USER SYSTEM



(57) Abstract: A method and system to input, update and verify personal information (e.g., contact information) is facilitated by a network-based Personal Information Management (PIM) service (214). A requesting user of the PIM service identifies one or more entities (e.g., personal contacts) for which the requesting user requires updated personal information (212). The PIM service responds by issuing an update request to each of the entities (e.g., an e-mail containing a URL link). The Web page may provide an input with various fields for receiving the personal information concerning an entity. Upon submission of said input, the PIM service updates a server-based personal information collection for the requesting user. A PIM client (12) may also contain a collection of personal information (18) which is then synchronized with the server-based collection.

**WO 01/33430 A1**

**BEST AVAILABLE COPY**



patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

**Published:**

— *With international search report.*

**METHOD AND SYSTEM FOR UPDATING USER INFORMATION MAINTAINED BY ANOTHER  
USER SYSTEM**

The present application claims the benefit of the filing date of U.S. provisional patent application no. 60/162,499, filed October 29, 1999 and U.S. utility patent application no. 09/566,053, filed May 5, 2000.

**FIELD OF THE INVENTION**

The present invention relates generally to the field of personal information management and, more specifically, to the publication, maintenance and synchronization of personal information, such as contact and address information, between multiple users connected to a network, such as the Internet.

**BACKGROUND OF THE INVENTION**

With the increasing movement towards the maintenance of personal information, such as personal address, contact and calendar information on personal computers (PCs) and Personal Digital Assistance (PDAs), the need to either maintain multiple, synchronized copies of such personal information, or have such personal information widely accessible, has increased. For example, a particular user may require his or her personal information to be stored on, and accessible from, a personal computer in the home environment, a personal computer in the work environment, a portable PDA or a mobile telephone that the user carries on his or her person. A user may also maintain a back-up copy of personal information on a diskette or at a remote location accessible via a network. For example, the company @Backup Corporation of San Diego, California, ([www.backup.com](http://www.backup.com)) provides the ability for a user to back-up a PC, which may store personal information, over the Internet.

Where a user has multiple devices each storing a local copy of personal information, it is desirable that all copies of the personal information stored on the various devices can be synchronized in an easy and convenient manner, as the management and maintenance of multiple copies of the personal information would otherwise become cumbersome. To this end, a number of software products are available that synchronize personal information stored by, for example, a Personal Information Manager (PIM) application resident on a computer system and a PDA. Specifically, Puma Technology, Inc. of San Jose, California, developed and markets the Intellisync software products that facilitate synchronization of personal information between a PDA (e.g., the Palm PDAs manufactured by 3Com Corporation of Santa Clara, California, or the numerous PDAs that operate under the Windows CE or PocketPC operating system developed by Microsoft Corporation of Redmond, Washington) and PC-based PIMs (e.g., Outlook developed by Microsoft Corporation or Lotus Notes distributed by International Business Machines of Armonk, New York). The Intellisync software typically requires that the PDA be connected to the computer system hosting the PIM, with synchronization between the local copies of the personal information being performed via a direct connection between the computer system and the PDA.

A recent service provided on the Internet is the storage and maintenance of personal information on a server, accessible via the Internet. A exemplary provider of such a service is PlanetAll.com ([www.planetall.com](http://www.planetall.com)) that allows a user to store personal information at a remote server, and to then have the user's personal information automatically included in the on-line address books of other users of the relevant service. The owner of the personal information is responsible for maintenance thereof, and the user may, simply by changing his or her personal information stored on the server, change the information that is viewable in the on-line address books of other users of the service. PlanetAll.com furthermore provides the ability to synchronize personal information maintained within a PIM (e.g., Microsoft Outlook) with

the personal information stored on the remote server. To this end, Intellisync for PlanetAll.com, developed by Puma Technology, Incorporated, is downloadable from the PlanetAll web site.

A number of web portals (e.g., Yahoo! Of Santa Clara, California and Excite Incorporated of Redwood City, California) have incorporated address book and calendering features into the services provided by these portals. For example, Yahoo! Incorporated provides Yahoo! Address Book, Yahoo! Calendar and Yahoo! To Do List services utilizing which a user can store address, calendar and to do information on a remote server operated by Yahoo! Incorporated. These web portals further offer synchronization software for free download from their respective web sites, this synchronization software providing the capability to synchronize copies of personal information stored on PDAs, PIMs and the remote server operated by the web portal. Both Yahoo! Incorporated and Excite Corporation offer the TrueSync synchronization software developed by Starfish, Incorporated of Scotts Valley, California.

Finally, eCode.com, Incorporated, offers web-based "business cards", whereby an "eCode" alias may be communicated to people, the alias being associated with personal information that has been accessible by the recipient of the alias from the web site operated by eCode.com, Inc. ([www.ecode.com](http://www.ecode.com)).

#### SUMMARY OF THE INVENTION

According to the present invention, there is provided a method including issuing, via a network, a request to a first user for input of personal information concerning the first user. The personal information concerning the first user is received via the network. A collection of personal information of a second user is then automatically updated with the personal information concerning the first user.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

**Figure 1** is a block diagram illustrating a network environment in which an exemplary embodiment of the present invention may be implemented.

**Figure 2** is a block diagram illustrating architectural details of an exemplary embodiment of a client services module of a client application, according to the present application.

**Figure 3** is a diagrammatic representation of communications between an exemplary client application and an application server.

**Figure 4** is a diagrammatic representation of a set of personal information fields, a number of which are selected to constitute a selection of virtual cards, according to an exemplary embodiment of the present invention.

**Figure 5** provides a diagrammatic representation of data structures containing information regarding a particular user, and a contact of that user, according to an exemplary embodiment.

**Figures 6A and 6B** are diagrammatic representations of an exemplary database structure that may be utilized to implement a server database.

**Figure 7** is a diagrammatic representation of an exemplary local database structure that may be maintained by a client application.

**Figure 8** illustrates an exemplary user interface to a personal information management application.

**Figure 9A** is a block diagram illustrating a personal information environment, according to an exemplary embodiment of the present invention.

**Figures 9B and 9C** are block diagrams illustrating respective exemplary installations by which a client application may interact with a network-based Personal Information Management (PIM) service to synchronize a server-based collection of personal information with one or more client-based collections of personal information.

**Figure 10** is an interaction diagram providing an overview of an exemplary method by which a user, who maintains a collection of personal information, may update a contact set of personal information.

**Figures 11A and 11B** show a flow chart illustrating a method, according to an exemplary embodiment of the present invention, of registering a user with a network-based Personal Information Management (PIM) service.

**Figures 12A and 12B** present a flow chart illustrating a method, according to an exemplary embodiment of the present invention, of inviting a contact, or a user, to provide new personal information to a user or to update or confirm existing personal information maintained by the user.

**Figure 13** illustrates an "exchange card" user interface, according to an exemplary embodiment of the present invention, that presents a number of input fields that receive personal and other information.

**Figure 14** illustrates an "preview message" interface, according to an exemplary embodiment of the present invention, that allows a requesting user to preview the contents and form of a request to be communicated to a potential new contact.

**Figure 15** illustrates a "confirmation message" interface, according to an exemplary embodiment of the present invention.

**Figure 16** illustrates an alternative embodiment of an "exchange card" interface that may be invoked responsive to a user action.

**Figure 17** is a flow chart illustrating a method, according to an exemplary embodiment of the present invention, of receiving personal information from a contact responsive to a request communicated from a requesting user.

**Figure 18** illustrates an exemplary message into which is embedded a URL.

**Figure 19** illustrates an exemplary "update user" interface that may be presented to receive updated personal information.

**Figure 20** illustrates a "login" interface, according to an exemplary embodiment of the present invention.

**Figure 21** illustrates a "confirmation" interface, according to an exemplary embodiment of the present invention.

**Figure 22** is a flow chart illustrating a method, according to an exemplary embodiment, of prompting a passive user to update active users

with personal information, according to an exemplary embodiment of the present invention.

Figure 23 is a diagrammatic representation of a machine in the exemplary form of a computer system within which a set of instructions, for causing the machine to perform any one of the methodologies of the present invention, may be executed.

#### DETAILED DESCRIPTION

A method and system to invite a contact to input, update and verify personal information maintained by a personal information manager (PIM) application are described. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details.

For the purposes of the present specification, the term "personal information" shall be taken to include, but not be limited to, address or contact information, calendar information, to do list information, financial information (e.g., credit card numbers), medical information or any other information specific to, and associated with, an individual or organization.

#### Architecture

Figure 1 is a block diagram illustrating a network environment 10 within which an exemplary embodiment of the present invention may be implemented. The network environment 10 is shown to include multiple client machines 12 that are coupled via a network 14 (e.g., the Internet) to a server farm 16. Each client machine 12 may host a client application 18, according to an exemplary embodiment of the present invention, that may function as a Personal Information Manager (PIM) and is responsible for the storage,

publishing, maintenance and synchronization of personal information concerning, for example, a user of the client machine. Each client machine 12 may be a personal computer (PC), a Personal Digital Assistant (PDA), a mobile telephone, a network appliance or any other machine capable of being coupled to a network and executing the client application 18.

The client machine 12 is furthermore shown to host a browser 20, such as the Internet Explorer browser developed by Microsoft Corporation, or the Netscape Navigator or Communicator browser developed by Netscape Communications, Incorporated of Mountain View, California. The client machine 12 may furthermore host a PIM 22, which may either be a stand-alone application (e.g., Microsoft Outlook) or part of a group-ware application (e.g., Lotus Notes). In an alternative embodiment of the present invention, the client application 18 may be fully integrated with and embodied within the PIM 22, or may itself may constitute a full-function PIM, and thus obviate the need for any further PIM 22.

The client application 18 is constituted by a front-end Graphical User Interface (GUI) 24 that, in an exemplary embodiment of the invention, may present a Windows® user interface where the client machine 12 is operated under a Windows 95/98/NT/2000 operating system developed by Microsoft Corporation. The GUI 24 provides a number of dialog boxes, information displays and interfaces for facilitating the convenient viewing, accessing, publishing and synchronization of personal information. The GUI 24 receives data from a client services module 26, which is accessed using Microsoft COM/DCOM technology.

The client services module 26 provides data services to both the GUI 24 and a local database 30. The client services module 26 is furthermore responsible for executing accesses to the local database 30 within which personal information regarding, for example, a user of the client machine 12 may be maintained. Specifically, the client services module 26 is responsible for the integrity and locking of the local database 30. In an exemplary

embodiment, the local database 30 comprises a lightweight object-oriented database developed by ObjectStore.

Components of the client services module 26 (including a synchronization engine 28) are responsible for synchronizing information maintained in the local database 30 with information maintained on a remote database accessible via the network 14. The client services module 26 communicates via a Secure Socket Layer (SSL) stack 27 over the network 14. Optionally, the client services module 26 also has the capability to synchronize with third party components hosted on, or coupled to, the client machine 12. For example, the client services module 26 may, via the synchronization engine 28, synchronize with the personal information manager (PIM) 22 (e.g., Microsoft Outlook or the Palm Desktop) or with a Personal Digital Assistant (PDA) 32 (e.g., the Palm Pilot by 3Com Corporation or any Windows CE device).

**Figure 1** also illustrates a mobile device 33, such as a PDA or cellular telephone, that is capable of establishing a wireless connection to the network 14, and thus to the server farm 16. The mobile device 33 hosts and executes a network access application, such as a Wireless Access Protocol (WAP) browser, that enables network-based interaction between the mobile device 33 and the server farm 16.

As discussed above, the client services module 26 of the client application 18 operates to synchronize the local database 30 with a remote database, such as a server database 34 maintained within the server farm 16. The server farm 16 is coupled to the network 14, and receives and transmits communications via a firewall 36, provided by a server farm provider, that implements well-known security features.

A resonate dispatch 38 may be hosted on a pair of Sun Ultra-SPARC machines, from Sun Microsystems of Mountain View, California. The resonate dispatch 38 performs load balancing operations between multiple machines on which an application server 40 and a web server 42 are hosted. In an

exemplary embodiment, both the application server 40 and the web server 42 may be hosted on a single Sun 450 machine from Sun Microsystems. The application server 40 may be developed utilizing Java technology developed by Sun Microsystems, and serve both the client services module 26 of the client application 18 on the client machine 12, and the web server 42. The application server 40 includes logic that allows a user, accessing the application server 40 via a client machine 12, to access only information for which the user has been granted permission. The application server 40 is furthermore responsible for sending personal information updates to the client services module 26 so as to synchronize the local database 30 with a specific subset of information maintained within the server database 34.

The application server 40 is also shown to embody an invitation module, in the exemplary form of an invitation servlet 41, that implements a number of invitation functions and methodologies whereby a person (or entity) may be invited to participate within a personal information management and maintenance service implemented via the server farm, or to update personal information regarding the person (or entity) that has been communicated to (or via) the server farm 16. Further information regarding the functioning of the invitation servlet 41 is provided below.

The web server 42 communicates with the resonant dispatch 38 via a SSL gateway 39 that encapsulates and unencapsulates Hypertext Transport Protocol (HTTP) communications issued from and to be received at the web server 42. The web server 42 may also be developed utilizing Java technology, and take advantage of the "Just In Time" (JIT) compiler and Sun Servlets. The application and web servers 42 and 40 provide full access to permitted data within the database 34 to a user of a remote client machine 12 by the browser 20. The application and web servers 42 and 40 further allow access to permitted data within the database 34 from any platform what provides web-browsing capabilities (e.g., client machines 12 operating under the Unix, Macintosh or Windows operating systems).

A Database Management System (DBMS) (also known as a data-mining server) 44 executes complex queries to the database 34 either when prompted or on a scheduled basis. The DBMS 44 may be hosted on a Sun Ultra Enterprise 4500 machine, while the server database 34 may be implemented using a RAID storage device. The server database 34 maintains synchronized copies of the local databases 30 that may be implemented on numerous client machines 12 coupled to the server farm 16, and accordingly the database 34, via the network 14. The server database 34 also records various permissions with respect to personal information by which personal information for a specific user may be accessible by, and accordingly published to, multiple other users. In effect, the server database 34 facilitates a system whereby, for example, an address book of a specific user (i.e., address information that is viewable by the specific user) is populated by information supplied and published by multiple other users. Accordingly, only a single copy of personal information concerning a specific user may exist within the server database 34, but this specific copy is accessible to multiple other users to whom an owner user has granted access permission. Further, the present invention envisages that the single copy of personal information for an owner user may be utilized to populate multiple local databases 30 maintained upon respective client machines 12. Accordingly, a local database 30 on a remote client machine 12 may be largely populated by information retrieved from the database 34, and which is maintained by an originator of such information.

#### Synchronization Engine and Synchronization Process

Figure 2 is a block diagram of an embodiment of the client services module 26, illustrating further architecture details thereof. The synchronization engine 28 is responsible for implementing two different synchronization processes, namely (1) a local synchronization operation wherein the client application 18, within which the module 26 is included, is synchronized with a local PIM 22 or a coupled PDA 32, and a (2) global (or

remote) synchronization wherein the client application 18 is synchronized with the application server 40, and the associated server database 34, via the network 14. The synchronization engine 28 furthermore has two modes of operation namely (1) an off-line mode wherein the client machine 12 is not connected to the network 14 and (2) an on-line mode wherein the client machine 12 is connected to the network 14. In the off-line mode, the synchronization engine 28 performs only local synchronization operations, which may be triggered at predetermined time intervals. Alternatively, a local synchronization operation may be triggered from the PIM 22 or from the PDA 32. When operating in the on-line mode, the synchronization engine 28 performs both local and global synchronization operations. Again, both these local and global synchronization operations may be initiated by the client application 18 at regular, periodic intervals, unless a synchronization operation is initiated externally, for example, from a PIM 22 or a PDA 32.

In one embodiment of the present invention, the client application 18 may detect when the client machine 12 establishes a connection to the network 14, and trigger a global synchronization operation responsive to the establishment of the connection. According to a further embodiment of the present invention, a "manual" synchronization operation is offered, whereby a user of the client machine 12 will be prompted to initiate a local and/or global synchronization operation.

During a synchronization operation, the GUI 24 interacts with the client services module 26 and the synchronization engine 28 to provide a textual and graphic display of the progress of a synchronization operation. For example, the GUI 24 may provide textual descriptions of operations being performed by the synchronization engine 28, and may also provide a progress bar showing the percentage of the synchronization operation that is complete, or that remains to be completed.

As illustrated in **Figure 2**, the client services module 26 includes the synchronization engine 28, a synchronization trader (application server) 52, an

eXtensible Markup Language (XML) stack 53, and a collection of other synchronization traders 54 and 56. The trader 52 is an object that resides in the synchronization engine's primary thread and manages all communication and interaction between the client application 18 and the application server 40. Specifically, the synchronization engine 28 polls the application server 40 for new messages (e.g., notifications of other user's subscriptions or updates) and will furthermore inform the application server 40 of new recruitment requests. The synchronization engine 28 manages all timed events for the client application 18, including calls to initiate synchronization with the application server 40 and database 34, as well as synchronization operations with external entities such as the PIM 22 or the PDA 32. The synchronization engine 28 furthermore includes an interface for communicating with the GUI 24, so as to facilitate the display of messages received from the application server 40, and the display of information concerning a synchronization operation.

The synchronization trader 52 is an object that is created by the synchronization engine 28 upon request from the GUI 24, or at specified time intervals. The synchronization trader 52 is responsible for managing all synchronization between the local database 30, the application server 40 and associated remote database 34. The synchronization traders 54 and 56 are similarly responsible for managing synchronization between the local database 30 and external entities such as the PIM 22 and the PDA 32. Each of these synchronization sources is represented within the synchronization engine by a respective synchronization trader 54 or 56. Each synchronization trader handles all data retrieval and update operations to and from the external entity (or source) such as the application server 40, the PIM 22 or the PDA 32.

The interfacing of the synchronization engine 28 with the GUI 24, the local database 30, the application server 40 and the PIM 22 will now be described. The synchronization engine 28 provides the GUI 24 (or any other client) with information from the application server 40. A portion of the functionality exported from the synchronization engine 28 is provided by a

Server Proxy Dynamic Link Library (DLL), while other functionality is provided by the synchronization traders 52, 54 or 56. Specifically, the synchronization engine 28 may implement an "automatic upgrade" function whereby the synchronization engine 28 automatically queries the application server 40 to determine whether an upgraded version of the client application 18 is available for upload to the client machine 12. The synchronization engine 28 further implements a "server session" functionality whereby a HTTP/TCP connection is established via the XML stack 53 (and the SSL stack 27) to the application server 40, and a number of methods are attempted to prompt the application server 40 to match contact information stored within local database 30 for which no copy exists within the server database 34. The client application 18 will then be notified of any matches that occur through messages from the application server 40 during a subsequent synchronization operation. Also included within the "server session" functionality is a contact match method, whereby personal information concerning the user of a specific client machine 12 may be published or communicated to further users.

The synchronization engine 28 may further implement a "message queue" functionality whereby pending messages held by the application server 40 are retrieved for processing and display by the client application 18, and a "recruiting" functionality.

As described above, the synchronization engine 28 manages all synchronization between external entities and the client application 18, and to this end obtains all updates from the local database 30 and from each of the synchronization traders 52, 54 and 56. If no conflicts arise, then both the external entity and the client application 18 will be updated with data from the other. The synchronization engine 28 will furthermore attempt to reconcile all conflicts that occur between data.

Each synchronization trader 52, 54 and 56 is responsible for exporting the database of the external entities that it represents as if it were compiled according to the scheme employed for the local database 30. Accordingly, each

of the synchronization traders 52, 54 and 56 is responsible for performing a mapping operation between fields of the local database 30, and a database maintained, for example, by the PIM 22 or the PDA 32. Each synchronization trader 52, 54 and 56 accesses an interface for updating the synchronization trader 52, 54 or 56 regarding any non-standard or user-defined fields that may be created within the local database 30.

#### Client-Server Protocol

One example of communications that may occur between the client application 18 and the application server 40 will now be described with reference to **Figure 3**, which provides a diagrammatic representation of communications between these two entities.

The client application 18 encodes information to be sent to the application server 40 in eXtensible Markup Language (XML), and propagates an XML stream over HTTP to the application server 40. As described above, the HTTP communications may further be encapsulated utilizing SSL to provide a higher degree of security. The client application 18 then waits for the application server's HTTP response, which is also an XML stream. The XML stream received by the client application 18 delivers C++ objects.

The client application 18 may have the application server 40 execute or perform several "functions". For example, the client application 18 may request the application server 40 authenticate a user. The instruction to the application server 40 to perform such functions requires that the client application 18 communicate several arguments to the application server 40. For example, when performing the above-mentioned authentication function, the client application 18 communicates a user name and a password to the application server 40. The application server 40 then returns a response, in the form of an authentication "cookie" in the case of a valid user authentication or an exception in the case of a failure.

**Figure 3** illustrates six exemplary functions that the client application 18 may request of the application server 40. Specifically, the client application may request an “authentication” function 60, a “get new contact identity” function 62, an “add new user” function 64, a “get new contacts updates” function 66, a “put contact updates” function 68 and a “close session” function 70. Each of the functions commences with a call from the client application 18 that includes the required arguments, and a response from the application server 40 that typically comprises an appropriate “cookie”.

The authentication function 60 requires the application server 40 to check and validate a user’s login name and password, responsive to which the application server 40 returns an authentication cookie to the client application 18 if the user is authenticated or an exception if the user is not authenticated. The client application 18 may then utilize the cookie for other function calls to identify the relevant user.

The “get new contact identity” function 62 is called by the client application 18 when new personal information (e.g., contact information) is added to the local database 30 of the client application 18. Responsive to the appropriate call, the application server 40 generates a new identification number that is then communicated to the client application 18 in a response from the application server 40.

The “add new user” function 64 adds a new user to the application server 40, and specifically to the database 34.

The “get new contacts updates” function 66 retrieves a list of personal information update operations that have been performed with respect to personal information stored on the database 34 subsequent to a previous synchronization operation between the client application 18 and the application server 40. To this end, the client application 18 communicates a sequence identifier, to be described in further detail below, to the application server 40 that performs a look-up of sequence identify numbers subsequent to the received sequence identifier to identify data operations that have occurred

since the previous synchronization operation. The application server 40 then responds by communicating messages detailing the updates that have occurred to the database 34 with respect to information to which the client application 18 has permissions.

The "put contact updates" function 68 is in effect the opposite of the "get new contacts updates" function 66, with the client application 18 communicating information concerning updates that have occurred with respect to the local database 30 to the application server 40. The application server 40 will then accordingly update the server database 34 with the received information, and propagate a response in the form of a sequence identify to the client application 18. It should be noted that a sequence identifier communicated from the client application 18 is for a sequence of operations with respect to the client application 18, whereas the sequence identifier communicated from the application server 40 to the client application 18 is with respect to a sequence of operations performed by the application server 40.

The "close session" function 70 essentially closes a session that has commenced with the performance of the "authentication" function 60, and accordingly disables or "kills" the authentication cookie maintained by the client application 18 for the current session.

#### Databases and Data Structures

An owning user may store a master set of fields of personal information concerning the owning user, and then designate different combinations and permutations of the fields of personal information as sub-sets of personal information. The owning user may publish a selected one or more of these sub-sets of personal information to a receiving user. The receiving user may then view the published sub-set as personal information, concerning the owning user, within a personal information repository (e.g., a PIM) of the receiving user. In one embodiment, the receiving user may populate, for

example, an address book utilizing a sub-set of personal information published to the receiving user by the owning user. Each of the published sub-sets of personal information concerning the owning user may be viewed as a calling card of the owning user, which may in turn be classified as a personal card, a business card or other cards for distribution and publication to multiple receiving users.

**Figure 4** is a high level, diagrammatic representation of the above described concept. Specifically, a master set 72 of personal information, comprising a number of fields 74, is defined, inputted and stored by an owning user. The input and storage of the master set 72 may, for example, be performed by a user via the client application 18, wherein the information is inputted via the GUI 24 and stored by the client services module 26 within the local database 30. The various fields 74 of personal information may include name, address, telephone, fax, e-mail, date, job title, work organization, medical, financial, family, interest, membership or any other personal information concerning the owning user.

The owning user may then record the designation of sub-sets of the information fields 74 as constituting respective virtual cards 78. By designating different sub-sets of fields 74 of the master set 72 as different virtual cards, or collections of fields, the owning user can define a collection 76 of virtual cards 78. For example, the owning user may define a first personal card that includes only a sub-set of information fields 74 that the owning user is willing to communicate to family members. The personal virtual card 78 may thus be designated as a "family" card. The owning user may then designate a second sub-set of information fields 74 as a "friends" virtual card 78, the relevant sub-set of information fields 74 comprising information that the owning user wishes to publish to friends. The owning user may then define a "business" virtual card 78 that encompasses a sub-set of information fields 74 that are appropriate for communication to a business client, colleague or associate.

Having then defined the collection 76 of virtual cards 78, the owning user may record the selection of one or more cards for publication to a selected receiving user (or subscriber). For example, the owning user may select the "family" virtual card 78 for publication to one or more family members, whereas the "business" virtual card 78 may be selected for publication to a number of business customers of the owning user.

Following the selection of predetermined "virtual" cards for publication to the receiving users, the sub-sets of fields 74 of personal information of the owning user are then published to respective receiving users in accordance with the selection of the appropriate virtual card. The operation of publishing the information to the receiving users may occur in a number of ways. An underlying premise is that the receiving user is granted permission to access the sub-set of fields 74 of personal information of the owning user embodied within the virtual card selected for publication to the receiving user. The access, by the receiving user, responsive to the granting of permission of access may occur in a number of ways. First, in a web-based system, a single remote database, such as the server database 34, may be maintained, with the receiving user being granted permission to view the sub-set of information fields 74 contained within the published virtual card as part of the receiving user's address book. In this case, only a single copy of at least the sub-set of personal information fields needs to be maintained on the server database 34.

In an alternative embodiment, the personal information of the owning user embodied in the published virtual card 78 may be utilized to publish a dedicated database of personal information that is accessible and owned by the receiving user. The dedicated database of personal information owned by the receiving user may, in one embodiment, be populated partially, or completely, by sub-sets of personal information (e.g., virtual cards) to which the receiving user has been granted access. The dedicated database may be maintained on a remote server or, as is the case in the network environment 10 illustrated in **Figure 1**, be maintained on a client machine 12 as the local database 30. In this

case the local database 30 is required to be periodically synchronized with the information that is published to the relevant receiving user within the server database 34.

In yet a further embodiment of the present invention, the server database 34 may be excluded from the publication operation, and a sub-set of personal information of a publishing user, maintained on a local database 30 of a first client machine 12, may be accessed by, or published to, a client application 18 residing on a further client machine 12 to either populate a local database 30 maintained on that further client machine 12, or to be viewed by an application hosted on that further client machine 12. In this situation, the local database 30 on the client machine 12 of the publishing user would in effect be functioning as a server database.

In light of the above, it will be appreciated that the network environment 10 illustrated in **Figure 1**, where local databases 30 maintained on different client machines 12 are synchronized via a server database 34 is merely one exemplary system by which the present invention may be implemented.

Referring again to the master set 72 of information fields 74 shown in **Figure 4**, it will be noted that a specific sub-set of information fields 74 is designated as default fields 80. The publishing user may designate certain information fields 74 as default fields 80, in which case such default fields may automatically be included within sub-sets of information fields allocated as comprising virtual cards 78 of a particular type. This is indicated in **Figure 4**, with the default fields 80 being included in each of the virtual cards 78 in the collection 76. A number of sets of default fields 80 may be defined for different virtual card types.

Dealing now with the exemplary embodiment of the present invention where a local database 30 of a client application 18 is maintained on a client machine 12, **Figure 5** provides an exemplary and high-level representation of information that may populate the local database 30. Specifically, the local database 30 may include the master set 72 of information fields 74 concerning

the owner of the local database (e.g., the publishing or owning user) of which certain fields 74 may be designated as default fields 80. In addition to the master set 72 of information concerning the owner user, the local database 30 may contain personal information concerning a non-owning user (hereinafter referred to as a "contact"). This information is illustrated in **Figure 5** as a set of contact information 82. The contact information 82 within the local database 30 is shown to comprise both a sub-set of published fields 84 and a further sub-set of unpublished fields 86. The sub-set of published fields 84 may be populated by information communicated to the client application 18 by the contact, for example in the form of a virtual card 78 shown in **Figure 4**. Accordingly, the contact, which is the publishing user with respect to the sub-set of published fields 84, generated the information that populates these published fields 84.

On the other hand, the sub-set of unpublished fields 86 is populated by information that may optionally be inputted into the local database 30 by the owner user. For example, the owner user may wish to record personal comments or information regarding the relevant contact. To this end, the owner user may wish to record a date at which the owner user last met the contact, and various circumstances of the meeting. This information would typically not be published by the relevant contact, but would be inputted and stored in the set of unpublished fields 86.

In summary, an exemplary local database 30 maintained by client application 18 and hosted on a client machine 12 may contain both owner user information (i.e., the master set 72 of information fields 74) and a number of sets of contact information 82. Each set of contact information 82 may in turn constitute a sub-set of published fields 84 and a further sub-set of unpublished fields 86.

Dealing now with the sub-set of published fields 84, in the case where the sub-set of published fields 84 is maintained in a local database 30, it is required that the information contained in these fields be periodically updated to conform to the information contained in corresponding fields of a master set

72 of personal information fields 74 maintained by the contact. This conforming operation is performed by periodically republishing the information from the source master set 72 to the sub-set of published fields 84.

In the exemplary embodiment of the present invention where local databases 30 are not maintained, and the sole information depository is the server database 34, the set of contact information 82 shown in **Figure 5** may be implemented as a permitted view, presented to a first user, of a selected sub-set of information fields 74 of a master set 72 of personal information of a second user. In this case, the second user is enabled to define the view of his or her information presented to the first user by assigning a specific and predefined virtual card 78 to the first user.

**Figures 6A and 6B** provide an entity-relationship diagram showing a database structure 90 that may be utilized to implement the server database 34 for an exemplary embodiment of the present invention. However, the database structure 90 may be implemented in either a server database 34 or a local database 30. In an exemplary embodiment of the present invention, the database structure 90 shown in **Figures 6A and 6B** is implemented within the server database 34, whereas a simplified database structure (described below) is implemented within each local database 30.

The database structure 90 is shown to include a number of tables, each of which includes a number of fields that are linked or keyed so as to implement a relational database.

The database structure 90 includes a users table 92 that maintains a respective record for each registered user. Each registered user may operate as both a publishing and a receiving (or target) user. The users table 92 records a user identifier, name, password, user type and last sequence identifier for each user. For example, a user type indication of "1" may indicate an active user, while a user type indication of "7" may indicate a passive user, as will be more fully described below. The last sequence identifier stored for each user record

indicates an operational sequence “peg” for an operation performed on a local database 30 with respect to the relevant user record.

The database structure 90 further includes a category\_fields table 94 and a category\_users table 96, the tables 94 and 96 together constituting a permission model 98, according to one exemplary embodiment of the present invention. The permission model 98 is utilized to define permission for particular fields of publishing user information to be published to a specific receiving user. The category\_fields table 94 maps information fields, defined in a fields table 100, to specific categories, in a many-to-many mapping, so that a single category may include multiple fields, and a single field may be included within multiple categories. In one embodiment of the present invention, categories are user-defined, and each category constitutes a sub-set of fields of personal information that may selectively be published by a publishing user to a receiving user. As such, the categories may be viewed as one exemplary mechanism by which to define a virtual card 78, with multiple categories comprising a collection 76 of virtual cards 78. The category\_fields table 94 includes a category identifier (cat\_id), a field identifier (field\_id), a user identifier (user\_id) and a sequence identifier (sequence\_id). The user identifier identifies the user (i.e., the publishing user) to which the relevant category-field mapping record belongs, while the sequence identifier again records an event sequence in which the creation of the relevant mapping occurred.

Each record within the category\_users table 96 includes an owner identifier (owner\_id) and a category identifier (category\_id) that records ownership of a particular category (e.g., a virtual card) by a specific user (e.g., the publishing user) for which a record exists within the users table 92. The owner identifier (owner\_id) within the category-users table 96 corresponds to a user identifier (user\_id) within the users table 92, which in turn corresponds to an owner identifier (owner\_id) within an ownership table 102.

Each record within the ownership table 102 further includes an owned identifier (owned\_id) that may be utilized to record ownership by a receiving

user of particular information regarding a publishing user. For example, where a receiving user supplements personal information regarding the publishing user within his or her local database, the owned identifier may be utilized to indicate such personal information concerning a first user that is "owned" by a second user.

A record within the ownership table 102 may further include a real user identifier (real\_user\_id) that indicates the identity of a first user that maintains information concerning that first user within the second user's local database 30. The real user identifier identifies information, concerning the "real user", that is owned and maintained by the "real user" but that populates another user's local database 30.

A current\_information table 104 stores actual values for each field (e.g., personal information field) for each user (that may comprise a contact) identified by the contact identifier (contact\_id).

As mentioned above, a record is maintained within the users table 92 for each registered publishing and receiving user within the network environment 10. However, a specific user may, within a local database 30 or on the server database 34, maintain a set of personal information records for an entity (e.g., a person or company) that is not a registered user. In this case, where the entity for which a record is maintained and owned is not a registered user, the entity (i.e., the non-registered user) is classified as a "contact" as opposed to a user. It will be appreciated that a user ID does not exist for the relevant contact. The taken\_contact\_id table 106 contains a record for each such contact, and maps a specific contact identifier (contact\_id) for the relevant contact to a user identifier (user\_id) to indicate ownership and origination of the relevant contact information.

It will furthermore be noted that an owned identifier (owned\_id) within the ownership table 102 may correspond to a contact identifier (contact\_id) within the taken\_contact\_id table 106.

A current\_information table 104 stores and maintains actual values for personal information fields for all contacts (some of which are registered users) whose information is stored within the database structure 90. Accordingly, each record within the current\_information table 104 includes a contact identifier (contact\_id), a field identifier (field\_id) and a field type (field\_type). The contact identifier (contact\_id) is shown to correspond to an owner identifier (owner\_id) in the category\_users table 96 in the case where the record in the table 104 is for information that is published, and therefore owned, by a registered user for which a record exists within the users table 92. On the other hand, where the record within the table 104 includes information that is not owned, or published, by a registered user (e.g., contact information regarding an entity (contact or user) that has been inputted into an address book and not in fact published to the address book), the contact identifier (contact\_id) correspond to a contact identifier within the table 106, which records a user, other than the entity to which the information pertains as an owner of the relevant contact identifier. It will also be noted that the contact identifier (contact\_id) in the table 106 in this case corresponds to the owner identifier (owned\_id) within the ownership table 102.

Multiple records for a single user may, in one embodiment, be stored within the current-information table, each record being authorized by a different entity. For example, multiple users may maintain separate records for a further single user.

In summary, the contact identifier (contact\_id) for each record within the current\_information table 104 corresponds either to an owner identifier (owner\_id) within the table 106 in the case where the information is published by an entity to which that information pertains, or corresponds to a contact identifier (contact\_id) within the taken\_contact\_id table 106 in the case where the information is not published by an entity to which the information pertains and that is manually included within a users contact information pertaining to the entity by the relevant user.

The field identifier (field\_id) for each record within the current\_information table 104 identifies a particular field corresponding to the field type identified within the same record. For example, the field type may comprise a telephone number, and the field identifier may identify the record as storing a home, work, or mobile telephone number.

A record within the current\_information table 104 also includes a value field, which stores an actual numeric, or alphanumeric, value for the relevant contact, identified by the contact identifier, for the field identified by the field identifier. For example, the value field would include an actual home telephone number.

A sequence identifier is also included within each record of the current\_information table 104 to identify an activity within an activity sequence by which the relevant record was updated or generated.

A period identifier (period\_id) included within each record of the current\_information table 104 provides a key to a record within periods\_dates table 108 that is populated with records indicating time periods during which a specific record within the current\_information table 104 is valid or extant. To this end, each record within the current\_information table 104 includes a period identifier (period\_id) to facilitate keying of a record to a record within the periods\_dates table 108 that includes a period name (period\_name), a start date (start\_dt), an end date (end\_dt) and a sequence identifier (sequence\_id). Consider, for example, the hypothesized situation where a particular user usually based in California spends a week in London, UK, and wishes to have his or her contact information updated for that period to reflect a London address and telephone number. In this situation, the user may identify a record within the current\_information table 104 as being valid for the duration of his or her stay in London by indicating appropriate start and end dates within a record within the periods\_dates table 108 that is keyed to a record within the current\_information table 104 that stores personal information (e.g., a work telephone number) that will be valid for the duration of the stay of the

user in London between the start and end dates. Further, where the record within the current\_information table 104 is for a contact (as opposed for a user), an owner of that contact information may pre-specify that an alternative set of personal contact information be valid and displayed for a particular period. In effect, by linking records within the current\_information table 104 to records within the periods\_dates table 108, a user may maintain different sets of personal information (e.g., contact information) that are published to receiving users over predetermined, specified time periods to reflect changes in the personal circumstances of the relevant publishing user. The period name may be utilized to attach a convenient and easily identified label to such time intervals. For example, a user could label a particular period as "visit to London", and attach the relevant time specification to a specific sub-set of personal information fields that is published to other receiving users.

The database structure 90 further includes a configuration table 110 that is populated by records indicating configuration information pertaining to, for example, the GUI 24 of a client application 18 hosted on a client machine 12. To this end, each record within the configuration table 110 includes a client identifier (client\_id) that identifies a particular client application 18 to which the configuration parameters are applicable.

A record within the configuration table 110 may furthermore be keyed to a user identifier (user\_id) thus making the configuration information stored in the relevant configuration record applicable to all client applications 18 for a particular user identified by the identifier. In this way, a user preference (e.g., a GUI display specification) specified via a particular client application 18 of a particular user can be uniformly applied across all client applications 18 associated with the relevant user and via which the user accesses information stored in the database structure 90. For example, a particular user may specify a particular configuration preference from a client application 18 executing on a computer system in a work environment. In this case, the configuration specification will also be communicated to a client application 18 that executes

on a computer system operating within a home environment of the same user. In this way, configuration preferences are applied to all client applications 18 via which a specific user accesses the database structure 90. The values indicating the configuration specification may be stored in the shown "path field".

The present invention also contemplates that users, for which user records exist within the users table 92, may attempt to recruit non-users to become registered with a personal information publication system supported by the database structure 90. To this end, the database structure 90 includes a recruitment table 112 that maintains a record of recruitment messages communicated from registered users to non-users. For example, a non-user may constitute a contact within a users address book, and the client application 18 may, in combination with the application server 40, provide a mechanism via which a user may invite a non-user to become a registered participant within the personal information publishing system. The recruitment table 112 maintains a record for each such "invitations" communicated from a user to a non-user, and attributes a recruiter identifier (recruiter\_id) to the sender of the invitation, a recruited identifier (recruited\_id) for each non-user who is successfully recruited and becomes registered as a user, a time record indicating the time at which the invitation was sent, and a record of the text of the invitation. A record is only created in the recruitment table 112 upon successful recruiting of a non-user as a user of the personal information publication system.

A blob table 114 provides a supplement to the current\_information table 104, and is utilized to store values for specific personal information fields where the values comprises anything beyond one line of continuous alphanumeric information. For example, where the stored personal information includes a carriage return, constitutes video, graphic or audio information, or other non-alphanumeric information, a value representative of this information may be stored within an appropriate record within the blob

table 114. To this end, the GUI 24 may support the display of a graphic image (e.g., a digital photograph in the JPEG format) of a contact that may be published to a receiving user. For example the GUI 24 may display a digital photograph in a manner so as to associate the digital photograph with other personal information pertaining to the publishing user on a display provided to the receiving user. In this case, a value that represents the digital photograph would be stored within an appropriate record within the blob table 114. Further, a particular publishing user may wish to publish a digitized audio recording of the correct pronunciation of his or her name. In this case, the client application 18, and specifically the GUI 24, may present a graphic icon or text that is user-selectable to invoke play back of the recorded digital audio pronunciation of the name published to the receiving user from the publishing user. In this case, a record within the blob table 114 would, in the value field, store an appropriate digitized representation of the audio recording. It is further envisaged that the blob table 114 may store records including values indicative of logos, or any other graphic, video or audio information that a particular may wish to include within his or her personal information to be published to receiving users.

Finally, the database structure 90 includes a sequence table 116 that maps each sequence identifier (sequence\_id) to a specific user identifier (user\_id), and a registration table 118 that records registration information for each registered user within a personal information publishing system. Specifically, a record for each registered user will be created within the registration table 118 upon the valid registration of the user, and specific registration information may be stored within a value field of each such registration record.

As mentioned above, the database structure 90 illustrated in **Figure 6** may be utilized to implement either a local database 30 or a server database 34. In an exemplary embodiment, it is envisaged that the database structure 90 be implemented within a server database 34, and a simplified version of this

database be mirrored by each local database 30. To this end, **Figure 7** illustrates an exemplary local database structure 120 comprising a number of information entities that may either constitute tables or, where the database constitutes an object database, information objects. Viewing the information entities as objects, the local database structure 120 includes a users object 122 that may store a sub-set of information stored in the users table 92 of the database structure 90. Specifically, the sub-set of stored information may comprise records for only those users that have elected to publish information to the receiving user that owns the local database 30 within which the structure 120 is implemented. The local database structure 120 may further include a contacts object 124 that corresponds substantially to the current\_information table 104 maintained within the server database 34. The contacts object 124 stores both published user information, and locally inputted contact information for entities whose records are maintained and viewed by a receiving user utilizing a local client application 18. Again, the records stored within the contacts object 124 may constitute only a sub-set of the records stored within the current\_information table 104, the sub-set being determined by permissions granted by publishing users to the receiving user, and also by information inputted locally by the receiving user.

The local database structure 120 may further include a categories object 126 that stores information corresponding approximately to information stored within the category\_fields table 94 and the category\_users table 96 within the database structure 90. Accordingly, the categories object 126 provides a local and user-specific version of the permission model 98 implemented by the category\_fields table 94 and the category\_users table 96 within the database structure 90.

The local database structure 120 finally includes an updates object 128 that maintains a record for each update to any of the tables 122-126, and accordingly stores a sequence identifier, data, table identifier, record identifier, field identifier, field index, value, source and previous update information for

each update that occurs within a local database 30. The previous update information stored within each record within the updates object 128 provides a pointer to a previous record that details a previous update. In this way, the client services module 26 of the client application 18 is able conveniently to trace a sequence (or history) of updates to a particular field of a particular record. This sequence, or history, of updates may then be communicated to the GUI 24 for display to a user.

#### User Interface and Methodology

In order to understand the methodologies implemented by a client application 18, and an application server 40, according to an exemplary embodiment, it is useful to provide a description of the displays generated by the GUI 24 of the client application 18 via which a user, in a publishing role, may input personal information concerning himself or herself, or information concerning a contact, and via which a user, in a publishing role, may elect to publish selected personal information, in the form of a virtual card 78, to receiving users. Furthermore, the GUI 24 facilitates the display to a user, within a receiving role, of personal information concerning other users that is published to the user in the receiving role. Further, the GUI 24 presents information concerning contacts to a user that the relevant user may have inputted him or herself.

**Figure 8** is a screen print showing a main window 130, according to an exemplary embodiment of the present invention, that may be generated by the GUI 24. While the UIs are described below as being Windows® UIs, it will be appreciated that such UIs may comprise markup language documents generated by a web server.

The main window 130 includes a tool bar 132, a “power find” panel 134 via which a user may conduct a search of contact information contained within the local database 30, a browser panel 136 that displays personal information

pertaining to contacts in the form of "contact cards" 138, a right panel 140, and a status bar 142.

Turning first to the "power find" panel 134, by inputting text into the text window provided in the "power find" panel 134, a user is able to search the local database 30. Specifically, the GUI 24 communicates an inputted search string to a thread-based fetch mechanism implemented in the client services module 26 that then returns search results to the GUI 24. The search results are displayed within the browser panel 136, and are refreshed every 0.5 seconds after each character input into the text window in the "power find" panel 134. In this way, the number of contacts located by the search is dynamically varied as the user inputs further characters into the search window. For example, after entering the leading letter "c", all contacts having a last name beginning with "c" will be displayed within the browser panel 136. Shortly after entering a subsequent "o" letter, only the contacts having a last name beginning with the letters "co" will be displayed following the 0.5 second dynamic refresh. Furthermore, the number of contacts located by current search parameters are displayed in the status bar 142.

The "power find" panel 134 further provides a "global search" option that is use-selectable to provide a more powerful searching tool, utilizing which the user may search multiple fields using respective criteria for each of those information fields.

The browser panel 136, as mentioned above, displays a virtual card for each contact of a selected category, or as located by a specific search query entered into the "power find" panel 134. A user may categorize various contacts for display purposes. For example, a user may designate a certain contact as being either a private contact, a business contact or as belonging to one or multiple other user-defined category. A tab, such as any one of those shown at 144, is then created for each user-defined category, and a user may conveniently view contact information for each respective category by performing a selection operation (e.g., utilizing a cursor control device such as

a mouse) on an appropriate tab for the relevant category. In the screen shot shown in **Figure 8**, the contacts in category "ALL" are shown in the browser panel 136.

Each contact card 138 shows a common design, and the information displayed in the contact cards within the browser panel 136 is not editable via the browser panel 136. Three default views for contact cards are provided. A first "photo" view provides merely a name caption and photograph, a "regular" view provides the photo and three to four customizable and user specifiable information fields, and a "full" view displays all the relevant contact's personal information fields stored within the local database 30.

As illustrated in **Figure 8**, the GUI 24 provides the capability for a user to perform a "drag-and-drop" operation with respect to a contact card 138. Specifically, by selecting, and then operating a cursor control device, a user can drag a contact information card to place the contact information card in a further category (e.g., by dragging the contact card to an appropriate tab 144, and then "releasing" the card) or to create a copy of the selected contact card as a virtual "memo" on a user interface desktop presented on a computer system. When placing a particular contact card in a further category utilizing a drag-and-drop operation, the user may furthermore specify whether the relevant contact card is to be moved to the further category, or to be copied to that further category.

Turning now to the right panel 140, this panel 140 includes a tab-set consisting of a "contact details" tab 146 associated with a contact details panel (not shown), a "permissions" tab 148 associated with a "permissions" panel (not shown) and an "on-line services" tab 150 associated with an on-line services panel (not shown).

The panel 140 further includes a "notes" section 158, within which the relevant user may record a personal note, or other information regarding the relevant contact.

Further, the panel 140 includes a "services" section 160 that displays information concerning the relevant contact that may be retrieved via the network 14, (e.g., the Internet) from an on-line publishing source. For example, the client services module 26 may issue a request to any one of a number of resources available on the Internet to obtain real-time, or near real-time, information pertaining to the relevant contact. For example, the client services module 26 may access the local database 30 to determine the residential city of the contact, and then formulate and propagate a request to an appropriate Internet service to retrieve weather and local time information for the relevant residential city. This information is then displayed, together with appropriate icons, as weather information 162 and local time information 164 within the services section 160 of the contact panel 140. Other real-time, near real-time, or on-line information, that may be presented within the panel 140 and retrieved based on personal information within the local database 30 regarding a particular contact, includes stock price information regarding an organization for which the contact works, "white page" information regarding an employer of the contract, a map indicating a home or work location for the contact, or any other on-line information that may be readily obtained utilizing personal information stored for the contact. On-line information as retrieved may then be displayed, together with the appropriate icons and graphics, within the services section 160 of the contact details panel 152.

Finally, the 140 also displays a digital image 166, for example stored in the blob table 114 within the database structure 90, for the relevant contact.

In the event that two or more contacts are selected within the browser panel 136, then the panel 140 displays the full names of the selected contacts as a list.

User-selection of the "permissions" tab 148 results in the permissions panel (not shown) being displayed in the right panel 140. Specifically, the permissions panel indicates the virtual card 78 of the relevant user that has been issued or published to the relevant contact. A particular user, in a publishing role, may, in one exemplary embodiment of the present invention,

publish a single virtual card (e.g., a business card) to a single contact. In an alternative embodiment, a publishing user may issue multiple virtual cards 78 to a single contact card in which case the various sub-sets of personal information embodied in each of these virtual business cards 78 may be combined to present a unified sub-set of the personal information of the publishing user to the receiving user (i.e., the contact).

The permissions panel further includes a “change this card” button, that is user-selectable to change the virtual card 78 assigned to the relevant contact.

**Figure 9A** is a block diagram illustrating a personal information environment 200, according to an exemplary embodiment of the present invention. The environment 200 includes a server machine 210 that interacts with a number of client machines 12 in a client-server relationship. In an alternative embodiment, the machines 12 may interact in peer-to-peer relationship.

In the exemplary personal information environment 200, a number of the client machines 12 are shown to execute the client application 18 as described above, each client application 18 executing to facilitate the maintenance of a respective collection 212 of personal information for a respective user. Each collection 212 of personal information may, for example, include contacts details (e.g., address, telephone, fax, pager and e-mail details) for a group of persons or entities.

The server machine 210, which forms part of the server farm 16 shown in **Figure 1**, is responsible for maintaining a global collection 214 of personal information, the global collection 214 being constituted by the respective collections 212 of a number of users. In a manner described above, the collections 212 maintained by the client applications 18 are duplicated within the global collection 214 by, for example, a synchronization process via the network 14. Such synchronization operations may be either client or server initiated.

Certain collections 212 of personal information (e.g., the illustrated user 4 collection) may be maintained exclusively by the server machine 210 within the context of the global collection 214, and need not necessarily be synchronized with a duplicate copy of the collection 212 as maintained on a client machine 12. In this case, the collection 212 of personal information may be viewed, accessed and updated via, for example, a browser application 20.

**Figure 9A** also illustrates that each collection 212 of personal information include an owner set 216 of personal information items and multiple user sets 218 of personal information. As described above, a particular user (e.g., user 1) may publish a selected subset of the owner set 216 to a collection 212 of a further user (e.g., user 2) to be incorporated as a user set 218 into the collection 212 of that further user. Updates to the owner set 216 of user 1 are then automatically propagated to, and included within, the user set 218 of the user 2. This may be referred to as a one-way linking of collections 212 of personal information. Similarly, the further user (e.g., user 2) may publish a selected subset of an owner set 216 to the collection 212 of user 1. In this case, a two-way linkage between the collections 212 of the relevant users (e.g., user 1 and user 2) is established. **Figure 9A** illustrates exemplary linkages between various collections 212 of user information for a number of users.

It will also be noted that a collection 212 of personal information may include a contact set 220 of personal information that is not populated out of the owner set 216 of another collection 212 within the global collection 214. For the purposes of differentiation, the term "user" shall refer to persons or entities that own a collection 212 of personal information in the context of the personal information environment 200. The term "contact" shall refer to persons or entities for which a user maintains information, but that does not themselves maintain a collection within the context of the personal information environment 200. For example, with reference to **Figure 9A**, the collection 212 for user 3 is shown to include contact sets 220 of personal information for two contacts (e.g., contact 1 and contact 2). The contact sets 220 are inputted by the

owner of the collection 212 (e.g., user 3) and reflect only information inputted by that user. For example, contacts 1 and 2 have no view of, or access to, the contact sets 220 of personal information concerning them.

It will be appreciated that a user set 218 of personal information will automatically be maintained through linkages to owner sets 216 of personal information that will be maintained by the respective owners. However, a contact set 220 may become outdated over time. Furthermore, a contact set 220 may be incomplete, as the user may only be aware of limited information regarding the relevant contact.

**Figures 9B and 9C** are block diagrams illustrating high-level exemplary installations by which a client application 18 may interact with a network-based Personal Information Management (PIM) service 8 to synchronize a server-based collection of personal information, stored within a server-side database, with one or more client-based collections of personal information, stored within client-side databases 30. Specifically, **Figure 9B** illustrates an exemplary embodiment wherein the client application 18 is a stand-alone application, and not integrated with a PIM 22 (e.g., Outlook). In this embodiment, the client application maintains its own collection of personal information within a client-side database 30. The client application 18 furthermore includes a synchronization engine that operates to synchronize the collection of personal information that it maintains with the server-side collection of personal information stored within the database 34. The synchronization engine furthermore operates to synchronize its collection of personal information with a collection of personal information maintained by the PIM 22.

In the exemplary embodiment illustrated in **Figure 9C**, the client application 18 is tightly integrated with, or included within, the PIM 22, and accordingly only a single client-side collection of personal information is maintained within a client-side database 30. Accordingly, the synchronization engine of the client application 18 operates only to synchronize a single client-

side collection of personal information with the server-side collection of personal information.

**Figure 10** is an interaction diagram providing an overview of an exemplary method 240 by which a user who maintains a collection 212 of personal information may update a contact set 220 of personal information, the contact set 220 comprising a record of personal information items concerning a contact. While the method 240 is described within the context of a personal information environment 200, such as that shown in **Figure 9A**, it will be appreciated that the discussed linkages between collections 212 and the publication of personal information through such linkages, is not essential to implementation of the present invention in a broad sense.

Referring specifically to **Figure 10**, at block 242, a user identifies a contact for which the user requires contact information, requires confirmation of existing contact information, or requires the updating of existing contact information. Such contact information is, in one embodiment, stored by the user as a contact set 220 within an owned collection 212 of personal information. The identification of the target contact may be through the input of an e-mail address, or other identifier by which the target contact may be identified.

At block 244, the target contact identifier is then communicated from the client machine 212 to the server machine 210, which may host the application server 40 discussed above with reference to **Figure 1**. The transmission at block 244 may be in the form of an e-mail, an HTTP/IP transmission, or any other network-based transmission.

At block 246, the invitation servlet 41, which forms part of the application server 40, executes to gather known personal information regarding the identified target contact. For example, a contact set 220 may be stored within a global collection 214 of personal information stored within the database 34. The contents of such a contact set 220 for the target contact may be retrieved at block 246. The known personal information may further, at

block 246, be embedded within a markup language document (e.g., an HTML document). More specifically, the invitation servlet 41 may issue a SQL request to the database management system 44 to retrieve the known personal information from the database 34. On receiving the requested personal information from the database management system 44, this personal information may be communicated to the web server 42 that populates a template with the dynamically retrieved information.

At block 248, a request for the input of personal information is issued from the server machine 210 by the network 14 to a contact (or target) client machine 12. The request for the input of personal information includes, in one embodiment, any known personal information concerning a contact associated with the contact client machine 12. The request for the input of personal information communicated at block 248 may include a request for the input of previously unknown data items (e.g., address and telephone data items), for the editing and updating of known personal information, or merely for the confirmation of known personal information. Accordingly, the term "input of personal information" shall, for the purposes of the present specification, be deemed to include the input of new information, the editing of known information, or the input of previously unknown information.

In one embodiment, the request communicated at block 248 comprises an electronic mail message to an e-mail address (or other network address) associated with the contact client machine 12, or with the contact. The request communicated at block 248 may further, in one embodiment, comprise a markup language document into which is embedded any known personal information. In an alternative embodiment, the request communicated at block 248 may include a location identifier (e.g., a Uniform Resource Locator (URL)) that identifies a network location at which a markup language document providing details of the request may be viewed by a contact. In this case, a further request/response operation (not shown) will be required to communicate the full request to the contact client machine 12.

In a further embodiment, at block 246, the invitation servlet 41 may also retrieve personal information regarding the requesting user from a relevant owner set 216 of personal information for the purposes of including this information in the request communicated to the contact at block 248. The personal information regarding the requesting user may accordingly be embedded within a markup language document included within the request, and may be offered as a *quid pro quo* to the contact to encourage the contact to respond to the request.

In one embodiment, the request communicated at block 248 in the form of a markup language document may present a number of labeled fields that prompt the contact for specified data items.

At block 250, the contact may then optionally input new personal information, for example, the labeled fields of a user interface generated responsive to receipt of the request. As described above, in one embodiment the request may be in the form of markup language document, or in other embodiments, the user interface into which the information is inputted at block 250 may be presented by a client application (e.g., a compiled executable program or a Java applet), that presents an interface specified at least partially in terms of the request.

At block 250, the contact may also edit and confirm known personal information. At block 252, the information inputted at block 250, as well as the information edited and confirmed, is communicated (e.g., as an HTTP PUT) from the contact client machine 12 to the server machine 210. Upon receiving the response, at block 254, the server machine 210, and more specifically the application server 40, parses the received response, extracts the information inputted by the contact at block 250, and updates the collection 212 of personal information for the user through the issuing of appropriate SQL requests to the DBMS 44. In this way, the user collection 212 of personal information maintained on the server machine 210, and specifically the contact set 220 of

personal information, is updated with information retrieved directly from the relevant contact.

At block 256, the updated contact set 220 of personal information is communicated from the server machine 210, via the network 14, to the user client machine 12. In one embodiment, this communication occurs as part of a synchronization operation between a user collection 212 of personal information maintained by the server machine 210 and a user collection 212 of personal information maintained by the client application 18 on the client machine 12. In this way, the personal information inputted, edited or confirmed at block 250 is disseminated from the server machine 210 to the user client machine 12.

In an alternative embodiment, the updated contact set 220 may be communicated from the server machine 210 to the user client machine 12 as an HTTP PUT message, or as an eXtensible Markup Language (XML) communication that is utilized by the client application 18 to update a local contact set 220 of personal information.

The method 240 discussed above with reference to **Figure 10** is advantageous in that, inter alia, the request communicated at block 248 to the contact includes known personal information, and thus reduces the effort required by the contact to respond to the request. Further, the inclusion of this known personal information may serve to verify the authenticity of the request and to serve as some level of assurance regarding the origin thereof.

A further advantage is that the newly inputted, edited or confirmed personal information is automatically propagated through from the server machine 210 to the client machine 12 to update a local copy of a contact set 220 of personal information maintained by a local client application 18 (e.g., a PIM such as Microsoft Outlook), and thus does not require that the requesting user manually attempt to update or synchronize the local personal (client-side) information with personal information hosted via the server machine 210.

Further details regarding a number of operations performed within the method 240 will now be described with reference to flow charts and exemplary user interfaces.

#### Methodology - Registration

Figures 11A and 11B show a flow chart illustrating a method 280, according to an exemplary embodiment of the present invention, of registering a user with a network-based Personal Information Management (PIM) service. The method 280 commences at block 282 with a registration operation by a potential user to register with a network-based PIM service, supported by the server farm 16 and the server machine 210. Specifically, the potential user may be prompted for an e-mail and password via a registration interface served up by the web server 42 via the network 14 to a browser 20 executing on a client machine 12.

At decision block 284, the user is then prompted by the PIM service to choose to enter personal information that will either be (1) a personal virtual card 78 or (2) a business virtual card 78. The prompting at decision block 284 is presented in conjunction with a dimmed display of an exemplary web client at display block 286. At block 288, a "create card" user interface (e.g., a markup language document) is displayed whereby a user can populate default personal or business card fields and select which fields are to be incorporated and displayed within the relevant personal or business card. The inputted virtual card information and virtual card layout are displayed at display block 290 for user confirmation.

At block 292, a Wireless Access Protocol (WAP) message is displayed to the user, informing the user about WAP capabilities offered by the PIM service, and prompting the user to accept or decline such a WAP service option.

At decision block 294, a determination is made as to whether the user accepted or rejected the WAP service option. In the event that the user accepted the WAP service option, the user is prompted for a phone number

and Personal Information Number (PIN) at block 296, presented with a confirmation screen at display block 298, and sent a confirmation and instruction electronic mail message block 300.

At block 302, the user is then presented with a synchronization (sync) marketing message advising the user regarding a personal information synchronization service offered by the network-based PIM service. The synchronization service, in one embodiment, synchronizes a collection 212 of personal information resident on a client machine 12 with, merely for example, such information maintained on a PDA 32, a mobile device 33 and by the PIM service.

The synchronization marketing message presented at block 312 also presents the user with the option of importing personal information, maintained on a client machine 12, into a user collection 212 maintained by the network-based PIM service (i.e., maintained on the server machine 210). This importing may initially be performed by a synchronization operation between the user collection 212 on the client machine and the user collection 212 on the server machine, the synchronization operation being performed by a synchronization agent (e.g., such as those offered by Starfish Incorporated or by Puma Technologies, Inc.).

At decision block 304, a determination is made as to whether the user elected to synchronize collections 212 of personal information maintained on the client machine 12 and the server machine 210. If so, at decision block 304, a determination is made as to whether the user has elected to download a "thin" or a "fat" version of the client application 18, the "thin" version offering reduced functionality relative to that offered by the "fat" client application. The "fat" client may, in one embodiment, offer enhanced synchronization functionality.

At blocks 308 and 310, the elected client application 18 is downloaded from the server machine 210, via the network 14, to the client machine 12, and at block 310, installed on the client machine 12.

At block 312, in view of the reduced synchronization capabilities of the "thin" client application 18, which may not perform a synchronization operation between a stand-alone PIM 20 (e.g., Microsoft Outlook), personal information (e.g., contact information) is exported from the PIM 20 and imported into the client application 18. This operation is not required for the "fat" client as the "fat" client application includes functionality to synchronize personal information maintained by the PIM 20 and the client application 18.

Returning to decision block 304, should it be determined that the user did not elect to download the client application 18 for the purposes of synchronizing a local collection 212 of personal information maintained on the client machine 12 with a network-based collection 212 of personal information maintained by the network-based PIM service, the user is prompted at decision block 320 to elect to send a virtual card (e.g., the personal or business virtual card) to a contact for the purposes of providing the contact with personal information regarding the user and for the purposes of harvesting, updating and confirming business information concerning the contact that may be known to the user. Should the user elect not to send such a virtual card, a network-based interface (e.g., a markup language document) that provides access to the network-based PIM service is presented at block 322.

On the other hand, should the user elect to send a virtual card to a contact, the user is prompted to input an e-mail address and name for the contact via an appropriate interface (e.g., a markup language document) at block 324, whereafter an invite operation is initiated at block 326.

#### Methodology - Issuing of Invite to a Contact to Update Personal Information

**Figures 12A and 12B** present a flow chart illustrating a method 340, according to an exemplary embodiment of the present invention, of inviting a contact, or user, to provide new personal information to a user, or to update or confirm existing personal information maintained by the user. The method 340, in one embodiment, is performed in a network-based environment with

the interfaces being server-generated (e.g., in the form of markup language documents) and being personalized by a client application (e.g., a browser). However, it will be appreciated that similar functionality may be achieved in a peer-to-peer, or distributed, network environment where multiple applications interact on a publish-and-subscribe basis. Accordingly, the client-server implementation described below is merely exemplary, and the present invention may be implemented within a peer-to-peer network architecture.

At block 342, the user may elect, via a network-based interface, to exchange virtual cards with a further entity (e.g., a contact or a user). At block 344, the PIM service 8 performs a check to determine whether the relevant user has defined one or more virtual cards for dissemination and exchange. If not, at block 346, a message prompts the user to create a virtual card, which is then created by the user, for example, in the manner described in co-pending U.S. patent application no. 09/566,053 utilizing the client application 18 via a network-based interface that is server-side generated.

At decision block 350, the user is prompted to identify the entity with which user wishes to exchange virtual cards as being an existing contact (e.g., a user or contact for which a record exists within the collection 212 of personal information of the user) or as being a new contact for which the user currently does not maintain a record within the collection 212.

Should the entity be identified as a new contact at decision block 350, at block 352, the user is prompted to input an e-mail address and name into a network-based interface.

At block 354, the user is prompted to select a virtual card, containing personal information regarding the user, to be communicated to the new contact as part of a request to provide personal information.

At decision block 356, the user may elect to preview the request (or invitation) to be communicated to the new contact, or add a personal message to default text to be included in the request.

Following a positive determination at decision block 356, at block 358, the PIM service 8 conducts a search to determine whether the new contact is an existing user of the PIM service 8 (e.g., whether a user record exists within the user table 92). If not, at display block 360, the preview message is displayed with the option to add text as the personal message.

On the other hand, should the new contact be an existing user of the PIM service 8, at decision block 362, the requesting user is queried for authorization to establish a link between a user collection 212 for the requesting user and a user collection 212 for the "new contact". In other words, the requesting user is prompted for authorization to publish selected information from the owner set 216 of the requesting user to a contact set 218 of personal information of the "new contact" user. Should the requesting user decline the link authorization at decision block 362, the method 340 terminates at block 364. On the other hand, should the requesting user authorize the link, linking functionality, as described for example in co-pending U.S. patent application no. 09/566,053 is initiated to establish a linkage, such as that described above, with reference to **Figure 9A**.

Returning to decision block 356, should the requesting user not elect to preview a request message to be communicated, or to add text as a personal message, at block 368 a search is conducted to determine whether the "new contact" is an existing user of the PIM service 8. If so, the method 340 proceeds to decision block 362, which is described above.

On the other hand, should the new contact not be identified as an existing user (e.g., no record exists within the user table 92), the method 30 proceeds to display block 370, where a confirmation message is displayed to the requesting user confirming that a request (or invitation), in an exemplary form of an electronic mail, was communicated to the contact.

At block 372, the user is prompted to select a category within which to categorize a record for the contact. Examples of such categories may include

personal contacts, business contacts, family members, and various groups or organizations which the contact is affiliated.

At block 374, an off-line record is created for the new contact in the form of a contact set 220 within the collection 212 of personal information for the user. The record, in one embodiment, is constructed by populating respective fields within the various tables of the database structure 90 illustrated in Figure 6.

At block 376, the network-based PIM service 8, via the server machine 210, issues a request (or invitation) to the contact to provide personal information. As described above, the request may include a URL, or other location identifier, that identifies a markup language document generated by the PIM service 8 (i.e., a network-based interface) that includes fields that may be populated with personal information of the new contact, and that also presents the new contact with specific information regarding the requested user.

At decision block 378, a determination is made as to whether the requesting user elected to exchange cards with another entity. If so, the method 340 loops back to decision block 350. Alternatively, following a negative determination at decision block 378, a "main screen" server-generated interface is displayed at block 380.

Returning to decision block 350, should the requesting user indicate that the contact with which the requesting user is to exchange virtual cards is an existing contact (e.g., the requesting user maintains a record within a collection 212 of personal information), the method 340 proceeds to block 382, where the requesting user is presented with a list of contacts, harvested from the collection 212. The requesting user may then select contacts with which he or she wishes to exchange virtual cards. The operations that follows block 382 substantially mirror those performed subsequent to block 352, as described above. A difference is that the request, or invitation, sent to the contact at block 376 differs in that the embedded URL points to an interface that, in addition to

presenting input fields for receiving the contact's personal information and that provides personal information regarding the requestor, also populates several of the input fields with known personal information regarding the contact. This known personal information regarding the contact may be harvested from a collection 212 maintained either on the client machine or by the PIM service 8 on the server machine 210.

#### User Interfaces - Issuing of Invitation/Requests From Requesting User

Figure 13 illustrates an "exchange card" user interface 450, according to an exemplary embodiment of the present invention, that presents a number of input fields to facilitate the operations described above with reference to Figures 12A and 12B.

The interface 450, which may comprise a markup language document (e.g., an HTML document) generated by the server machine 210 of the PIM service 8, includes name input fields 452 and an e-mail input field 454 via which name and e-mail address for a contact may be entered by the requesting user at block 352. A card selection field 456 presents a drop-down menu of virtual card types that a requesting user may select to communicate to a new contact at block 354. A card window 458 displays the contents of a virtual card selected in the field 456, and displays an image 460 and other virtual card data items 462 that are included within the relevant virtual card. The data items 462 may include any of the data items, or information fields, discussed above with reference to Figure 8 or discussed in copending U.S. patent application serial no. 09/566,053.

A user selection of a "create new card" button 461 invokes card creation functionality. An "personal message" window 463 allows a requesting user to input personalized text to be included within a request to be communicated to the new contact.

Hypertext 464 is user-selectable to present a list of existing contacts to the requesting user in an interface using which the requesting user may select contacts to which requests are sent.

Finally, a "preview message" button 466 is user-selectable to invoke a "preview message" interface 470.

**Figure 14** illustrates a "preview message" interface 470, according to an exemplary embodiment of the present invention, that allows the requesting user to preview the contents and form of a request to be communicated, on behalf of the requesting user, to the new contact by the network-based PIM service 8. As discussed above, the request is shown to include a URL 472 that identifies a server-generated interface for presenting and receiving personal information regarding the contact to which the request is communicated. The preview message interface 470 also displays a card window 458 that is embedded within the request, the card window 458 again displaying a selected virtual card that presents personal information regarding the requesting user to the new contact.

An "add personal message" 474 allows the requesting user to input a personal message which is then included within the request at the location 476.

A "send" button is user-selectable to authorize the transmission of the request (or invitation) as displayed within the preview message interface 470.

**Figure 15** illustrates a "confirmation message" interface 480, according to an exemplary embodiment of the present invention, that may be displayed at block 370, described with reference to **Figures 12A and 12B**. Again, the interface 480 may be a markup language document generated by the PIM servers 8 accessed by a browser 20 executing on a client machine.

The interface 480 provides confirmation that a request (e.g., an e-mail message) was communicated to the identified contact, or contacts. Further, the interface 480 provides a "file as" input field 482 that allows the user to specify the format under which the new contact is to be filed within the collection 212 of the requesting user. A "category" field 484 allows the requesting user to

specify one of a set of predetermined categories in which to categorize the record with the new contact within the collection 212.

An "exchange more cards" button 486 is user-selectable to issue requests to further contacts, and a "finish" button 488 is user-selectable to terminate the request and invitation process.

**Figure 16** provides an alternative embodiment of an "exchange card" interface 490 that may be invoked responsive to user-selection of the hypertext 464 of interface 440 shown in **Figure 13**. The interface 490 is similar to the interface 450 shown in **Figure 13**, but differs in that, as opposed to requiring that the requesting user input name and e-mail address details, a list of existing contacts is presented in a table 492 for user selection (e.g., by checking the check boxes 494). Each of the selected existing contacts then receives a request, in the manner requested above.

#### Methodology - Contact Input of Personal Information

**Figure 17** is a flow chart illustrating a method 500, according to an exemplary embodiment of the present invention, of receiving personal information from a contact responsive to a request communicated from a requesting user.

At block 502, having received a request in an exemplary form of an electronic mail message, a contact user may select the URL link embedded within the URL message. **Figure 18** illustrates an exemplary message 504 that shows an exemplary URL 506.

Returning to **Figure 17**, at block 508, a user interface, in the form of a web page or other markup language document, that includes a form to receive updated personal information is presented within a browser 20 of the contact. The web page presented at block 508 is, in one embodiment, generated by the server machine 210 of the network-based PIM service 8. **Figure 19** illustrates an exemplary "update user" interface 510 that may be presented at block 508,

the interface 510 including an update form 513 to receive the updated personal information.

At decision block 512, the contact is prompted to identify him or herself as an existing user by, merely for example, selection of the "existing user" button 514 presented within the interface 510. In the event that the contact is identified as an existing user, at decision block 512, the contact is prompted for authorization to establish a link with the requesting user. If the contact declines authorization, the method 500 terminates at block 518. Alternatively, should the contact authorize linking, the contact is prompted to log in at block 520, following which a link operation, as described above, and as described in co-pending U.S. patent application serial no. 09/566,053, is initiated at block 522. **Figure 20** illustrates an exemplary "login" interface 524 that may be presented at block 520.

Returning to decision block 512, if the contact is identified as not being an existing user, at block 526, a determination is made as to whether the relevant contact has previously provided personal information to an existing user of the network-based PIM service 8. If so, the contact may, in one embodiment, be recorded as a "type 7" user (i.e., a passive user) within the user table 92 illustrated in **Figure 6**.

If the contact is not an existing passive user, at block 528, the update form 513 is displayed with known personal information included within appropriate fields of the form. In one embodiment, the fields displayed within the form 513 for receiving personal information may be any one of the fields discussed above, or discussed in copending U.S. patent application serial no. 09/566,053. In a further embodiment, only fields for data items corresponding to the data items 462 that the user has selected to be published to the contact are included within the update form 513. In this way, a *quid pro quo* exchange of personal information may be facilitated. For example, where the new contact is a business acquaintance of a user, the user will communicate a virtual business card to the contact. The information that the user deems appropriate

to be communicated to a business contact will accordingly be displayed within the virtual card data items 462. Accordingly, the request issued by the user for personal information from the contact will include corresponding fields which are appropriate, in the users view, to be requested from a business contact. On the other hand, where the virtual card published by the user to the contact is a personal virtual card, more extensive information may be published to, and accordingly requested from, the further contact.

Referring to **Figure 19**, it will also be noted that certain of the fields within the update form 513 are populated with personal information that is already known to the contact. In this way, the burden of inputting personal information placed on the contact is somewhat reduced.

At block 530, the contact may then input personal information into the update form 513 of the interface 510, and submit this information to the network-based PIM service 8 by user-selection of the "update" button 531 presented within the interface 510.

At block 532, the network-based PIM service 8 creates a new "type 7" (or passive user) record within the user's table 92.

At block 534, a confirmation interface 536, an exemplary embodiment of which is shown in **Figure 21**, is displayed to the contact and presents to the contact a virtual card and marketing message. Referring specifically to **Figure 21**, the user interface 536 is shown to include two marketing messages 540, a constructed virtual card 538, and options 542 that are user-selectable to prompt the contact regarding certain function options provided by the network-based PIM service 8.

Returning to block 526, should it be determined that the contact is in fact an existing, passive user, the method 500 branches to display block 544. As mentioned above, a passive user may be defined as a user that has previously submitted personal information to a user of the network-based PIM service 8 in response to a request, but is in fact not an active, registered user (e.g., a type 1 user) of the network-based PIM service 8. In the case where the contact is

identified as a passive users, the contact is, within the context of the update interface 510, presented with a history of the user to which the passive user has previously provided contact information. Such a listing of further users is illustrated at 546 in **Figure 19**. Selecting one of the list of users allows a contact to exercise an "auto form fill" function, whereby the information previously presented to a selected user is utilized automatically to populate the update form 513. In this way, the contact is spared the inconvenience of having to re-input the same information responsive to requests from multiple users, but at the same time is not required to engage as an active user of the network-based PIM service 8. The operations following block 544 are the operations performed at blocks 528 and 530, as described above.

At decision block 550, a determination is made as to whether to register the contact, based on the options 542 selected within the user interface 536. For example, should a contact indicate that he or she wishes automatically to receive updates from the user, this indication is facilitated by registering the contact as an active (e.g., a type 1) user within the network-based PIM service 8. Accordingly, following a positive determination at decision block 550, a web registration operation is commenced at block 552. An example of such a web registration operation is provided by the method 280 described above with reference to **Figures 11A and 11B**.

On the other hand, following a negative determination at decision block 550, at block 554, the PIM service 8 issues an update to the sending user (e.g., by way of the synchronization operation discussed above with reference to **Figure 10**). At decision block 556, the requesting user may either accept or reject the personal information changes or updates submitted by the contact. If the updates are accepted, at block 558, the record for the contact is confirmed and incorporated into the collection 212 of personal information for the requesting user.

#### Updating of Passive User Details

**Figure 22** is a flow chart illustrating a method 560, according to an exemplary embodiment of the present invention, of prompting a passive user (e.g., a type 7 user) to communicate updated personal information to one or more active users (e.g., type 1 users) of the network-based PIM service 8. The method 560 is implemented utilizing the components of the network-based PIM service 8 discussed above, and utilizing interfaces similar to those presented in **Figures 18 - 21**.

At block 560, an active user A 572 sends a request, in the manner discussed above, to a passive user B 574 for personal information. At block 564, the passive user B 574 responds to the request with personal current information, and elects not to "join" the network-based PIM service 8 and accordingly to remain a passive user (e.g., a type 7 user).

At some future time, at block 566, a further active user C 576 sends a further invitation to join the PIM service 8 and/or a request for personal information to the passive user B 574. The invitation and/or request communicated at block 566 may be substantially similar to the invitation and/or request communicated from the active user A 572 at block 562. The receipt of multiple invitations and/or requests by a passive user B 574 may, for example, occur where multiple active users maintain personal information for the relevant passive user B 574.

At block 568, the passive user B again responds to the invitation and/or request with current personal information and elects not to join the PIM service 8 and to remain a passive user.

At block 570, the PIM service 8, and specifically the invitation server 241, may recognize that the passive user B 574 has now provided current personal information to an active user of the service 8, and has also provided personal information to one or more other active users at a previous time. It may or may not be that the personal information most recently provided by the passive user B 574 to an active user is more current, and may reflect changes from information previously provided to other active users. Accordingly,

responsive to the detection of the provision of updated personal information by a specific passive user, the service 8 may present the passive user with the option to update personal information, concerning the passive user, that has previously been provided to active users. For example, in the exemplary situation depicted in **Figure 22**, should the passive user C 576 elect to update all previous users, the user information for passive user B 574, maintained by the PIM service for the active user A 572, would automatically be updated with the personal information that the passive user B 574 has supplied to the active user C 576 at block 568.

The method 560 is advantageous in that a "passive" user of the PIM service 8 is provided with a convenient mechanism whereby current personal information can be published to all active users of the PIM service 8 when updated personal information is provided to a single active user. Thus, the passive user, even though not an active participant or publisher within the PIM service 8, can be provided with some of the personal information management functionality provided by the PIM service 8.

### Computer System

**Figure 23** shows a diagrammatic representation of machine in the exemplary form of a computer system 600 within which a set of instructions, for causing the machine to perform any one of the methodologies discussed above, may be executed. In alternative embodiments, the machine may comprise a network router, a network switch, a network bridge, Personal Digital Assistant (PDA), a cellular telephone, a web appliance or any machine capable of executing a sequence of instructions that specify actions to be taken by that machine.

The computer system 600 includes a processor 602, a main memory 604 and a static memory 606, which communicate with each other via a bus 608. The computer system 600 may further include a video display unit 610 (e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT)). The computer

system 600 also includes an alpha-numeric input device 612 (e.g. a keyboard), a cursor control device 614 (e.g. a mouse), a disk drive unit 616, a signal generation device 618 (e.g. a speaker) and a network interface device 620.

The disk drive unit 616 includes a machine-readable medium 622 on which is stored a set of instructions (i.e., software) 624 embodying any one, or all, of the methodologies described above. The software 624 is also shown to reside, completely or at least partially, within the main memory 604 and/or within the processor 602. The software 624 may further be transmitted or received via the network interface device 620. For the purposes of this specification, the term "machine-readable medium" shall be taken to include any medium which is capable of storing or encoding a sequence of instructions for execution by the machine and that cause the machine to perform any one of the methodologies of the present invention. The term "machine-readable medium" shall accordingly be taken to include, but not be limited to, solid-state memories, optical and magnetic disks, and carrier wave signals.

Thus, a method and system to input, update and verify personal information facilitated by a network-based Personal Information Management (PIM) service have been described. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method including:

issuing, via a network, a request to a first user for input of personal information concerning the first user;

receiving, via the network, the personal information concerning the first user; and

automatically updating a collection of personal information of a second user with the personal information concerning the first user.

2. The method of claim 1 wherein the issuing of the request comprises generating user interface information to display a user interface presented on a display device of a remote computer coupled to the network, the user interface information specifying a plurality of data fields to receive the personal information concerning the first user.
3. The method of claim 2 wherein the user interface information comprises a markup language document.
4. The method of claim 3 wherein the request comprises an electronic mail message incorporating a location identifier for the markup language document.
5. The method of claim 4 wherein the location identifier comprises a Uniform Resource Locator (URL).

6. The method of claim 1 wherein the issuing of the request comprises including known personal information concerning the first user within the request, the request being for confirmation by the first user of the known personal information concerning the first user.
7. The method of claim 6 wherein the known personal information is included within a markup language document presented by the request.
8. The method of claim 7 wherein the markup language document is presented by the request as a hypertext link to the markup language document.
9. The method of claim 1 wherein the request is issued from a server device to a first remote network device associated with the first user, the server device maintaining the collection of personal information of the second user.
10. The method of claim 9 wherein the issuing of the request comprises including known personal information concerning the first user in the request, the known personal information being extracted from the collection of personal information of the second user maintained by the server device.
11. The method of claim 9 including updating a further collection of personal information of the second user maintained on a second remote network device, coupled to the network, with the personal information concerning the first user.
12. The method of claim 11 wherein the updating of the further collection comprises performing a synchronization operation between the collection of personal information maintained by the server device and the further collection of personal information maintained by the second remote device, the synchronization operation being performed via the network.

13. The method of claim 1 wherein the request is issued from a second remote network device associated with the second user to a first remote network device associated with the first user, the second remote network device maintaining the collection of personal information of the second user.
14. The method of claim 13 wherein the issuing of the request comprises including known personal information concerning the first user within the request and wherein the known personal information is extracted from the collection of personal information of the second user maintained by the second remote network device.
15. The method of claim 1 including receiving, at a server device, an identification of the first user from the second user, determining whether the first user has published first personal information concerning the first user via the server device and, if so, facilitating a link request to the first user to publish the first personal information concerning the first user to the second user.
16. The method of claim 15 wherein the facilitating comprises publishing second personal information concerning the second user to the first user as part of the link request to the first user.
17. The method of claim 16 wherein the link request comprises an offer to establish a link between the collection of personal information of the first user and the published second information concerning the second user so that the published second information is automatically incorporated into the collection of personal information of the first user.
18. The method of claim 1 wherein the request includes personal information concerning the second user.

19. The method of claim 19 wherein the personal information concerning the second user included within the request comprises a first set of data items selected by the second user.
20. The method of claim 19 wherein the request for the input of the personal information concerning the first user is for the input of a second set of data items comprising the personal information regarding the first user, the second set of data items corresponding to the first set of data items comprising the personal information concerning the second user.
21. The method of claim 1 including determining whether the first user has provided first personal information responsive to a further request from a third user to the first user for input of personal information and, if so, then including in the request an option, available to the first user, to automatically provide the first personal information responsive to the request.
22. The method of claim 21 wherein the inclusion of the option includes identifying the third user in the request.
23. The method of claim 22 including, responsive to user selection of an identifier identifying the third party, displaying the first personal information to the first user.
24. A machine-readable medium storing a sequence of instructions that, when executed by a machine, cause the machine to:

issue, via a network, a request to a first user for input of personal information concerning the first user;

receive, via the network, the personal information concerning the first user; and

automatically update a collection of personal information of a second user with the personal information concerning the first user.

25. A system including:

an invitation module to issue, via a network, a request to a first user for input of personal information concerning the first user and to receive, via the network, the personal information concerning the first user; and

a database manager automatically to update a collection of personal information of a second user with the personal information concerning the first user.

26. The system of claim 25 wherein the invitation module is to generate user interface information to display a user interface presented on a display device of a remote computer coupled to the network, the user interface information specifying a plurality of data fields to receive the personal information concerning the first user.

27. The system of claim 26 wherein the user interface information comprises a markup language document.

28. The system of claim 27 wherein the invitation module is to issue the request as an electronic mail message incorporating a location identifier for the markup language document.

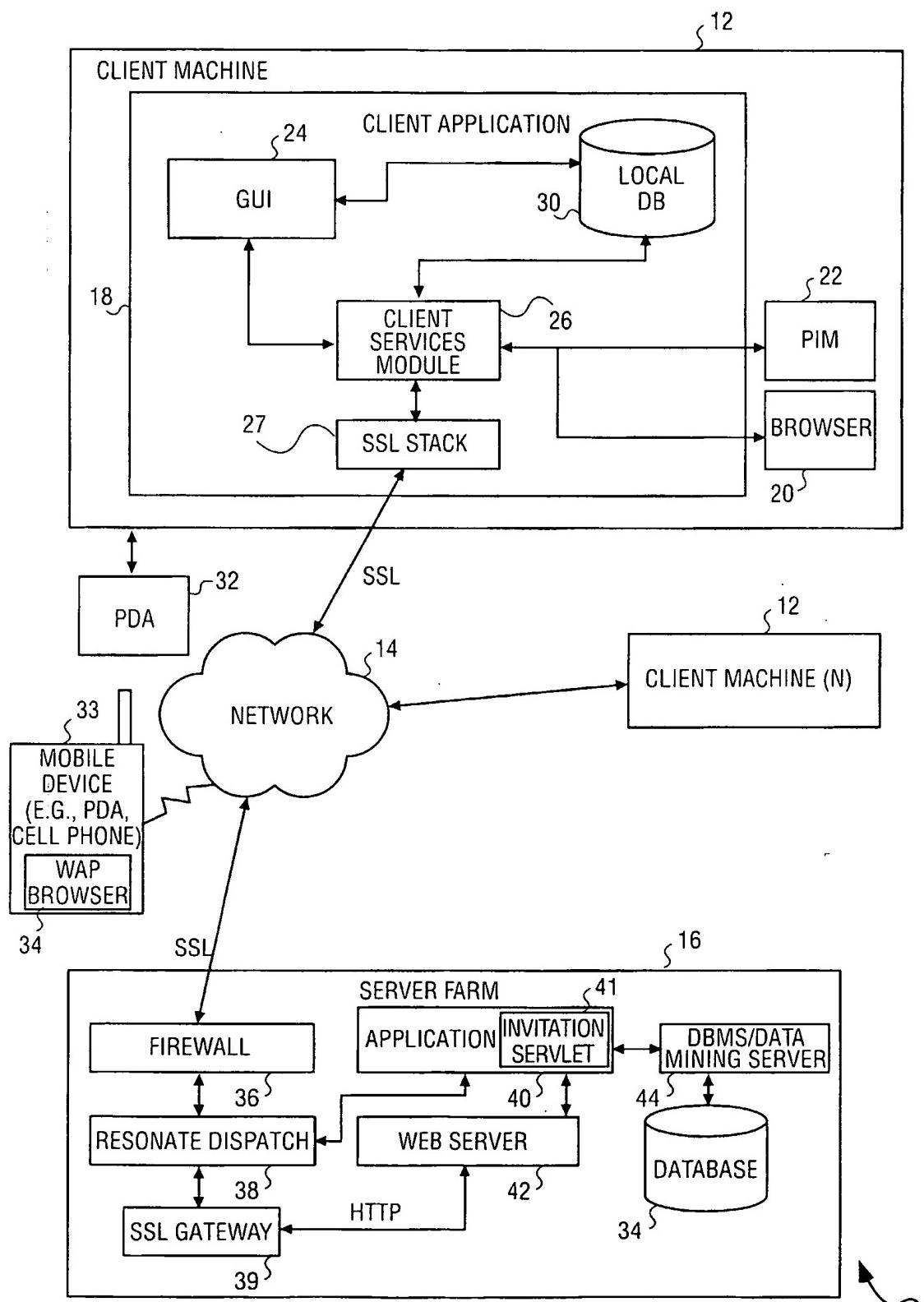
29. The system of claim 28 wherein the location identifier comprises a Uniform Resource Locator (URL).
30. The system of claim 25 wherein the invitation module is to issue the request including known personal information concerning the first user within the request, the request being for confirmation by the first user of the known personal information concerning the first user.
31. The system of claim 30 wherein the known personal information is included within a markup language document presented by the request.
32. The system of claim 30 wherein the markup language document is presented by the request as a hypertext link to the markup language document.
33. The system of claim 25 wherein the invitation module is to issue the request from a server device to a first remote network device associated with the first user, the server device maintaining the collection of personal information of the second user.
34. The system of claim 33 wherein the invitation module is to issue the request to include known personal information concerning the first user in the request, the known personal information being extracted by the invitation module from the collection of personal information of the second user maintained by the server device.
35. The system of claim 33 including a synchronization engine to update a further collection of personal information of the second user maintained on a second remote network device, coupled to the network, with the personal information concerning the first user.

36. The system of claim 35 wherein the synchronization engine is to update the further collection by performing a synchronization operation between the collection of personal information maintained by the server device and the further collection of personal information maintained by the second remote device, the synchronization operation being performed via the network.

37. A system including:

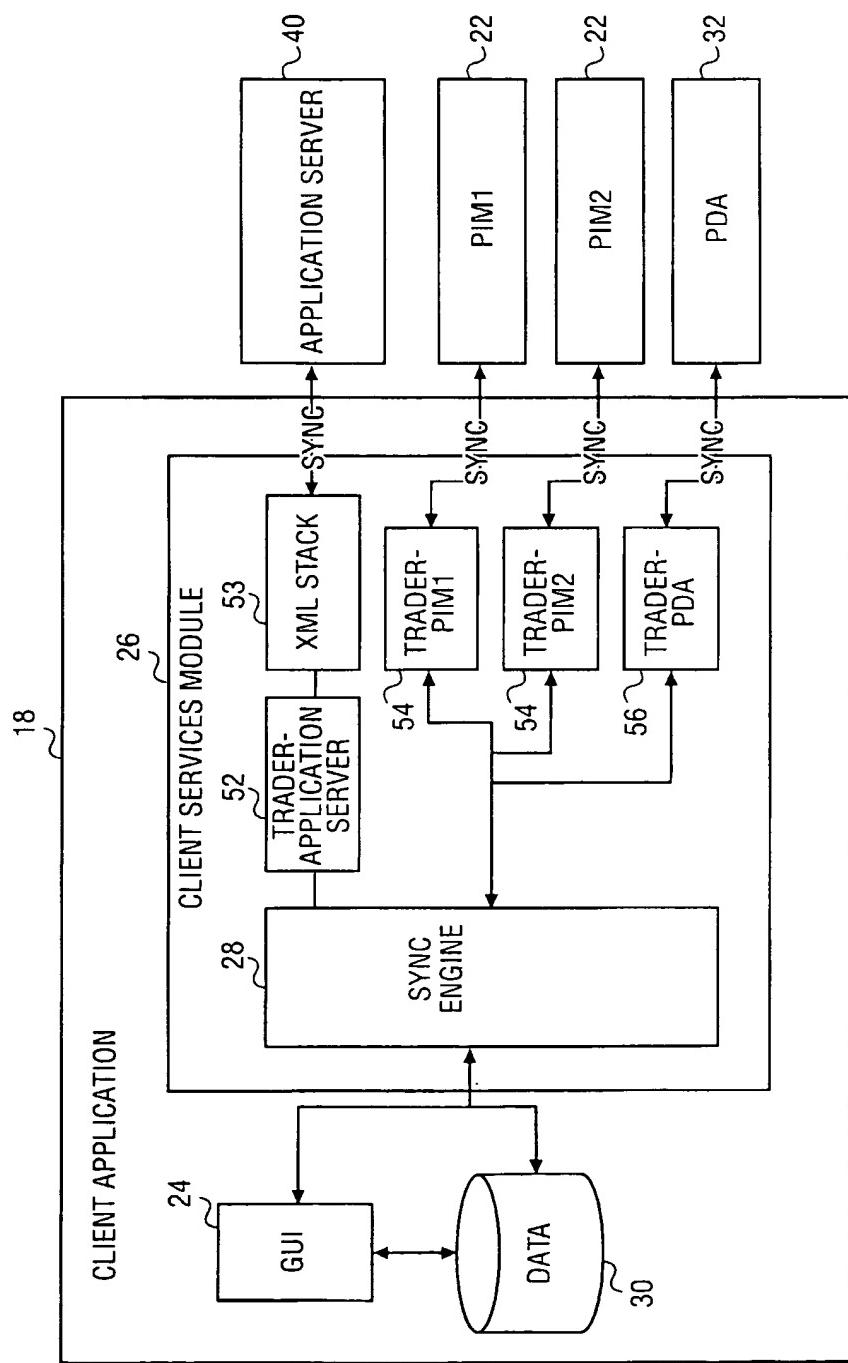
first means for issuing, via a network, a request to a first user for input of personal information concerning the first user and for receiving, via the network, the personal information concerning the first user; and

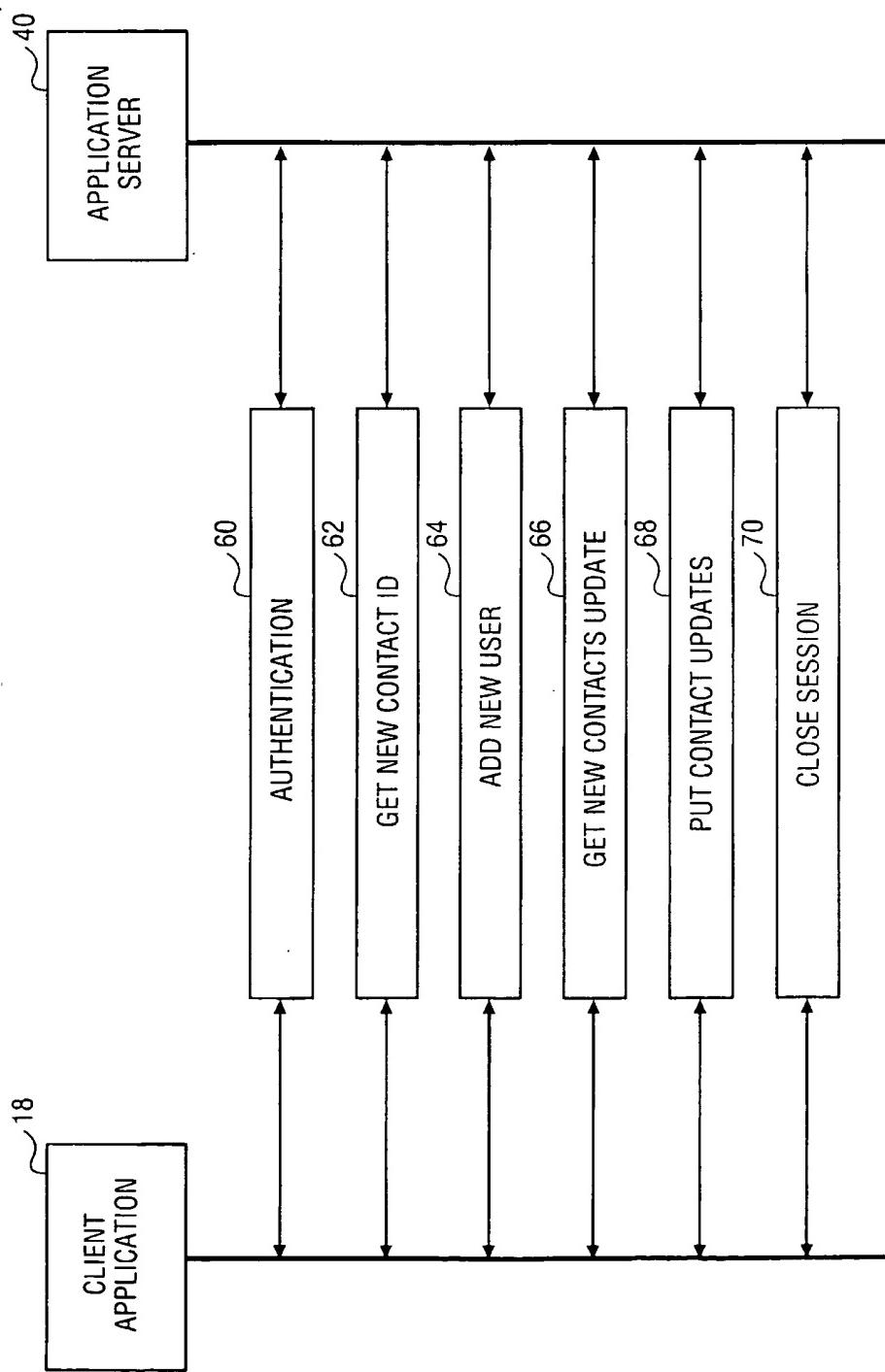
second means for automatically updating a collection of personal information of a second user with the personal information concerning the first user.

**FIG. 1**

SUBSTITUTE SHEET (RULE 26)

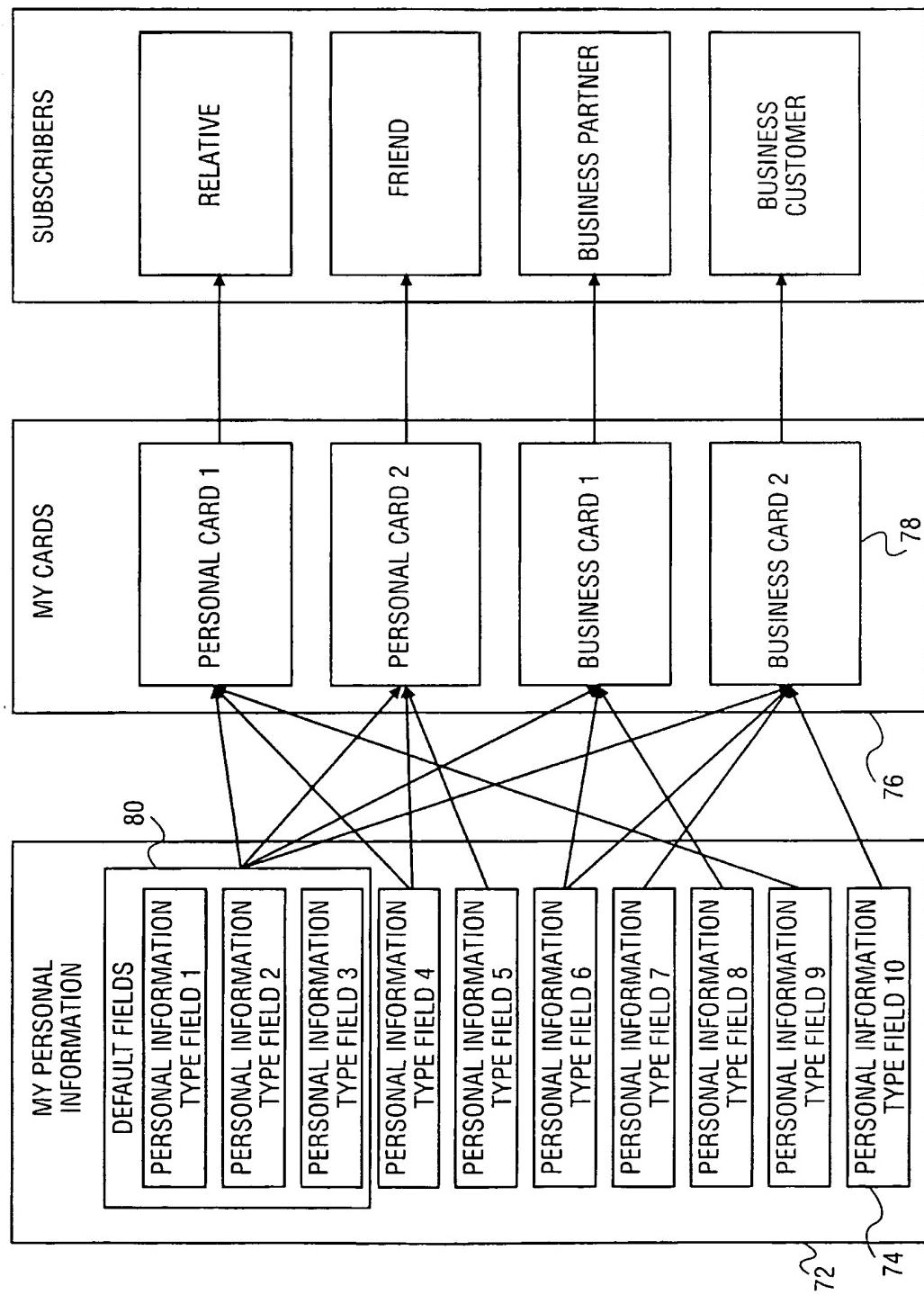
FIG. 2



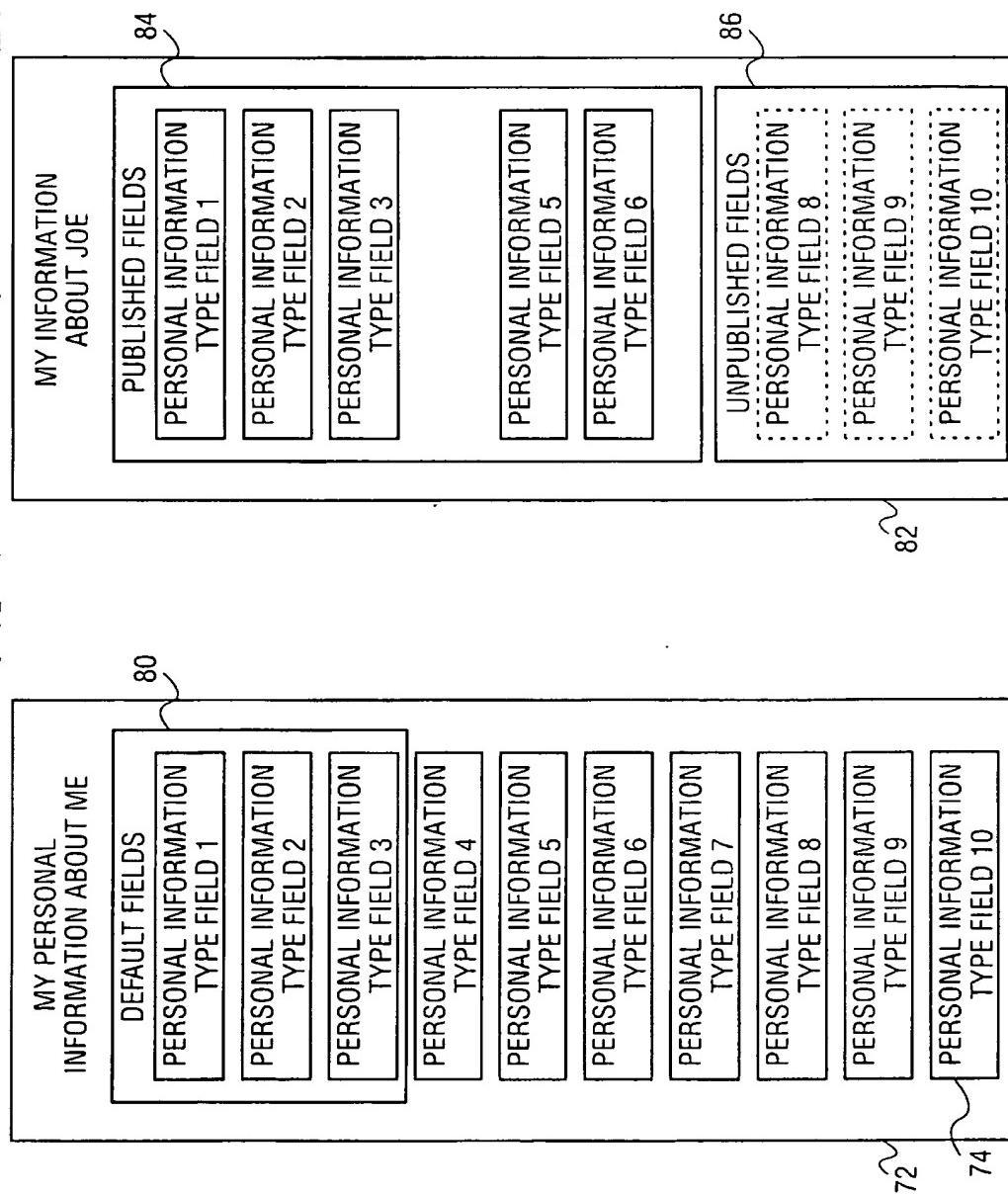
**FIG. 3**

4/27

FIG. 4



5/27

**FIG. 5**

6/27

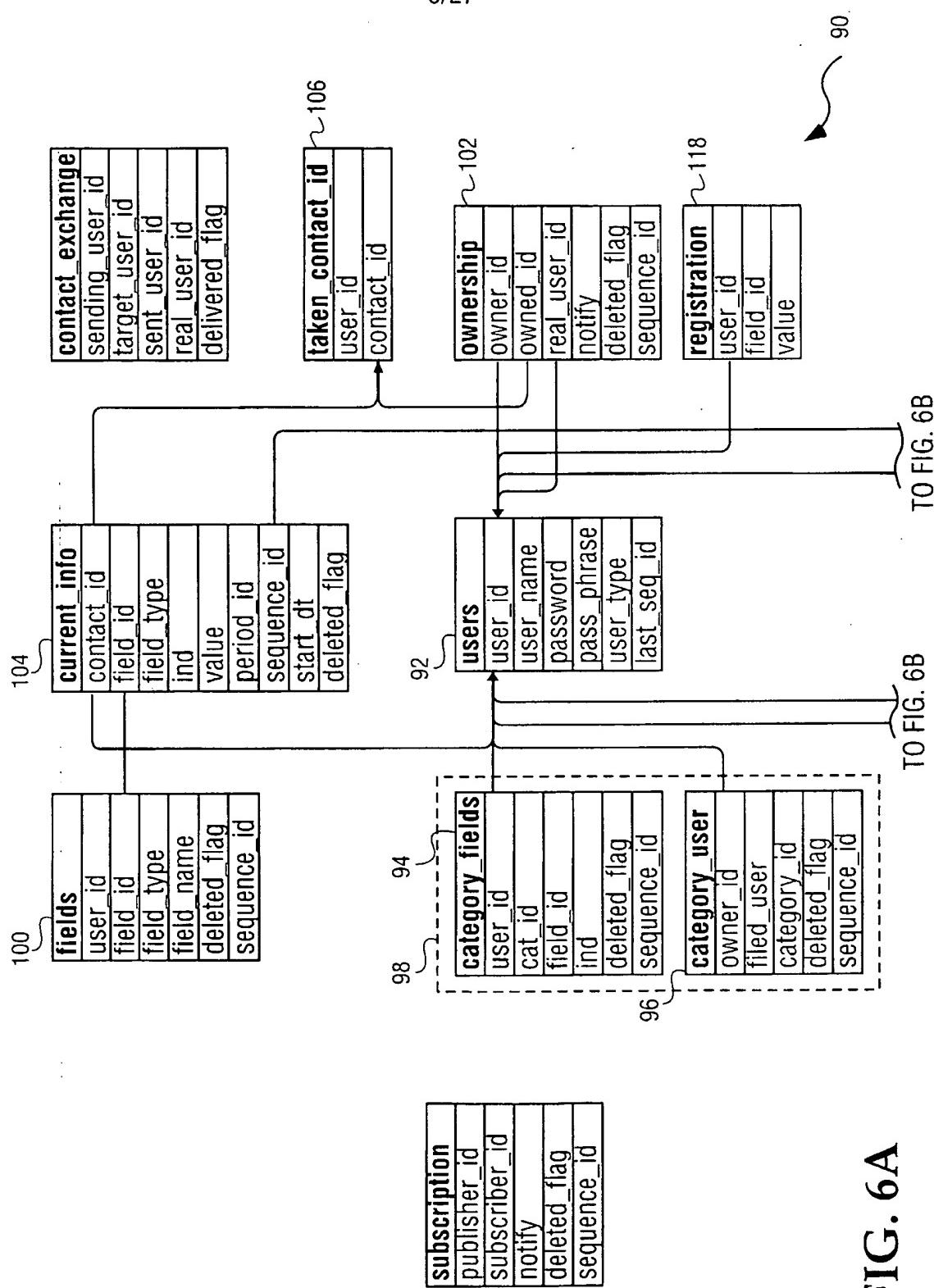


FIG. 6A

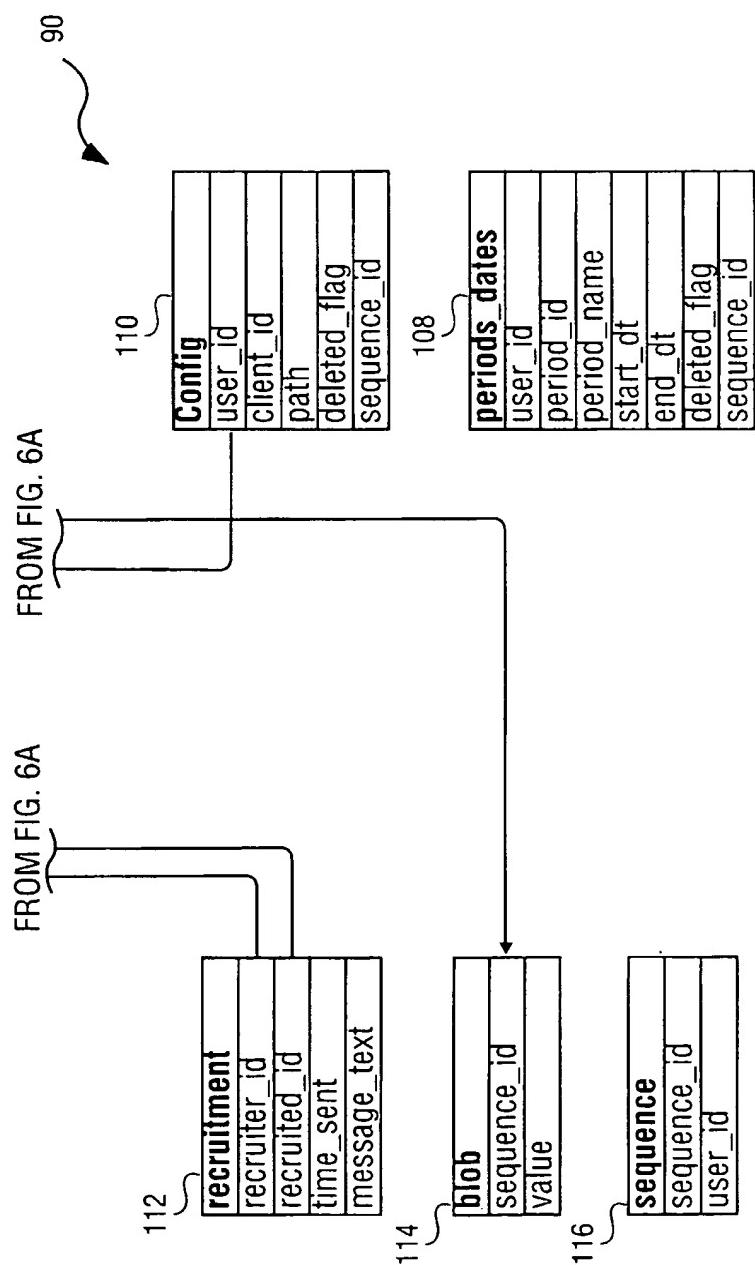
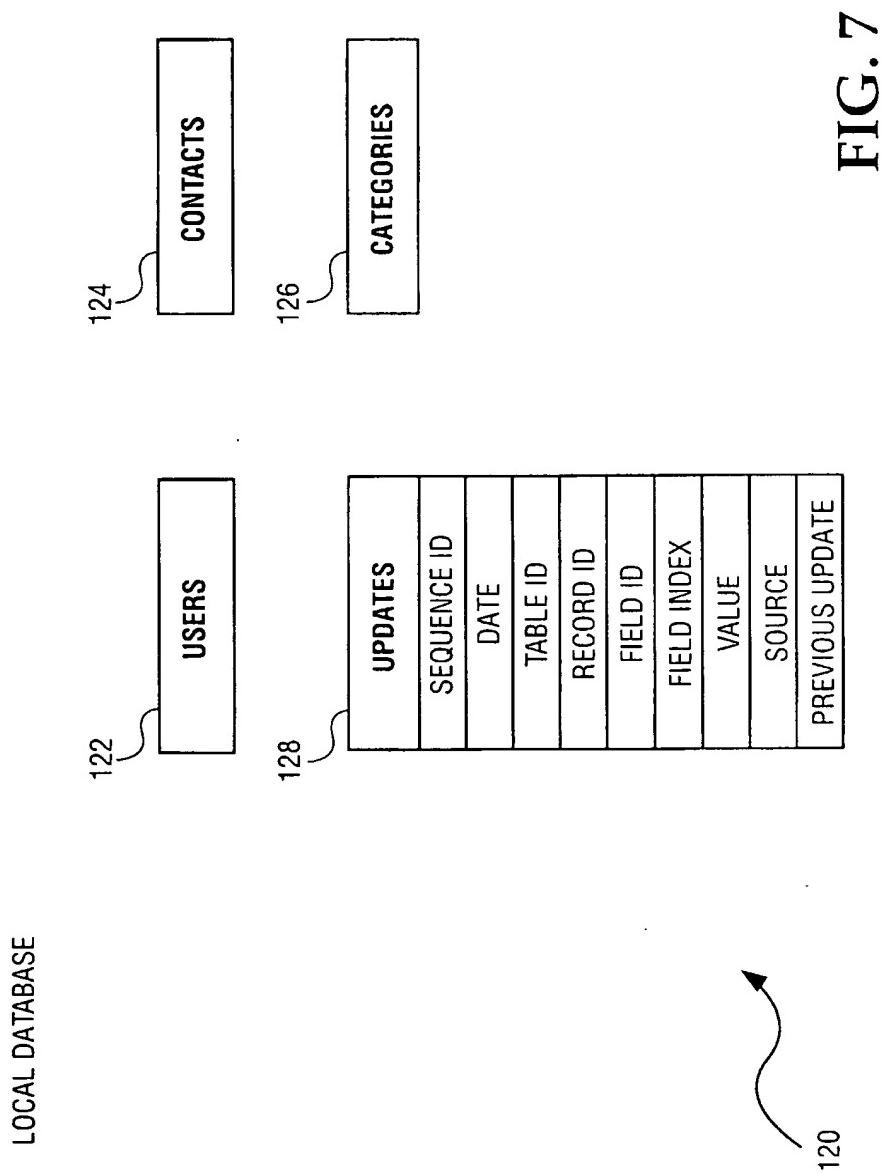
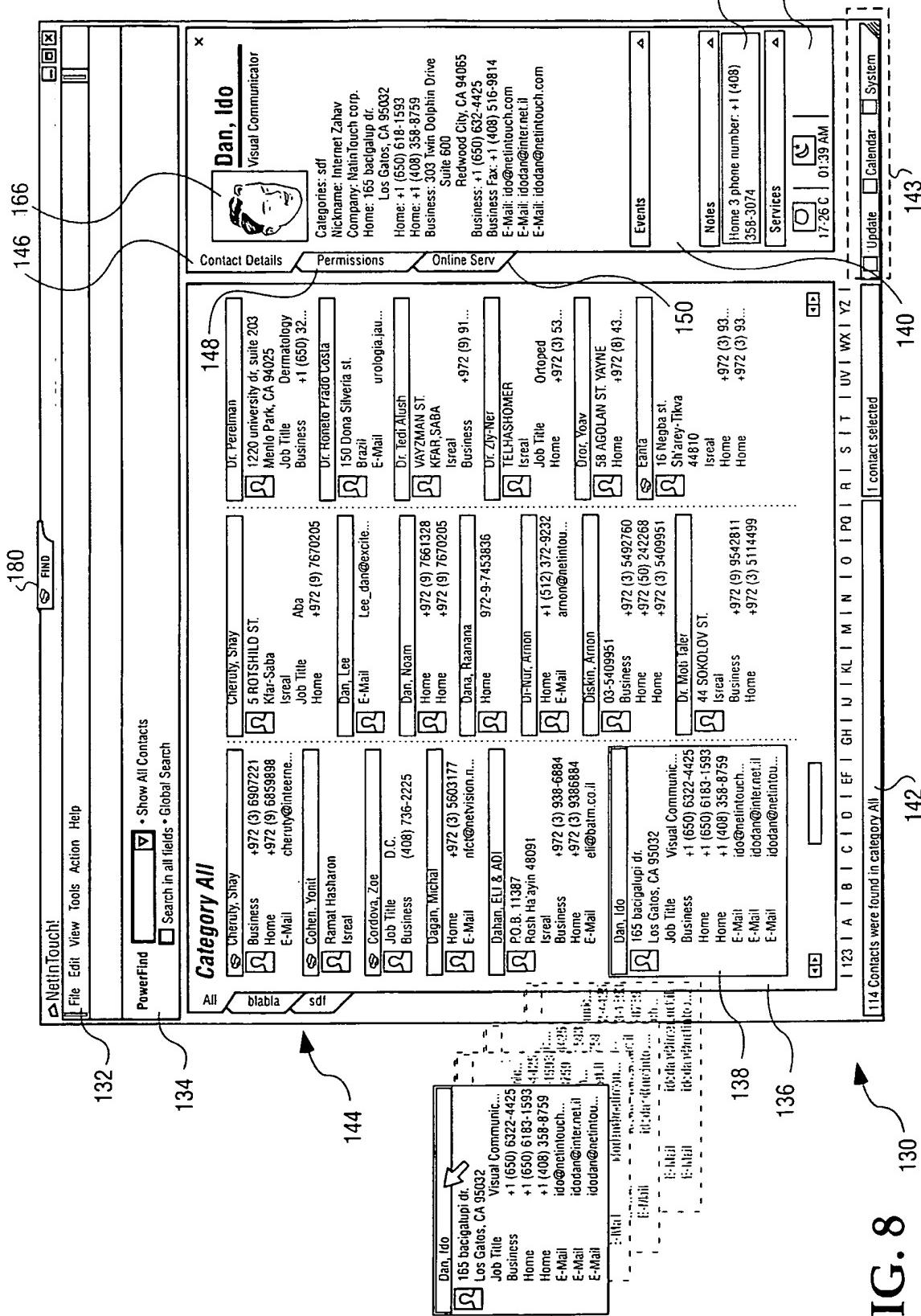


FIG. 6B

8/27

**FIG. 7**

9/27



**SUBSTITUTE SHEET (RULE 26)**

10/27

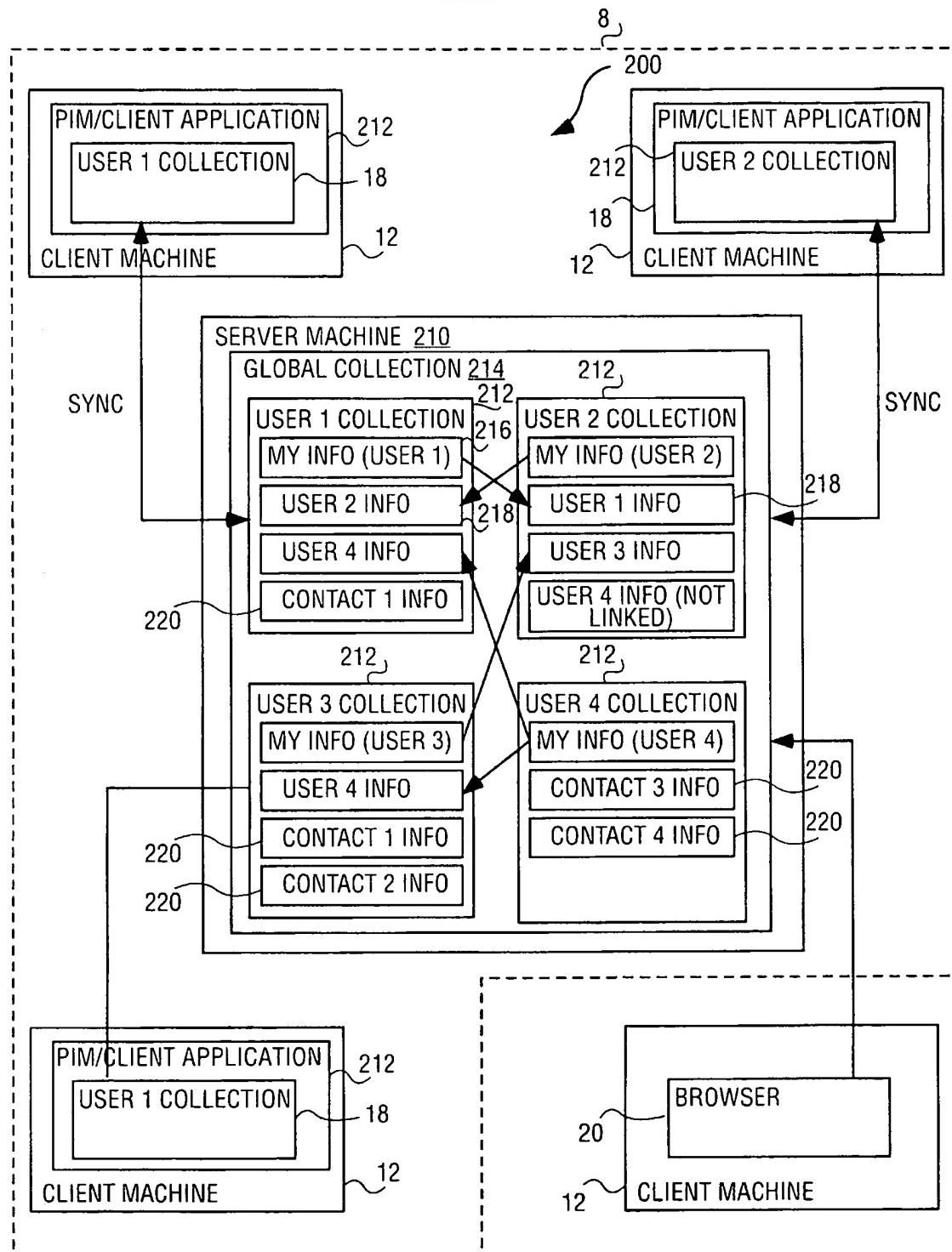


FIG. 9A

11/27

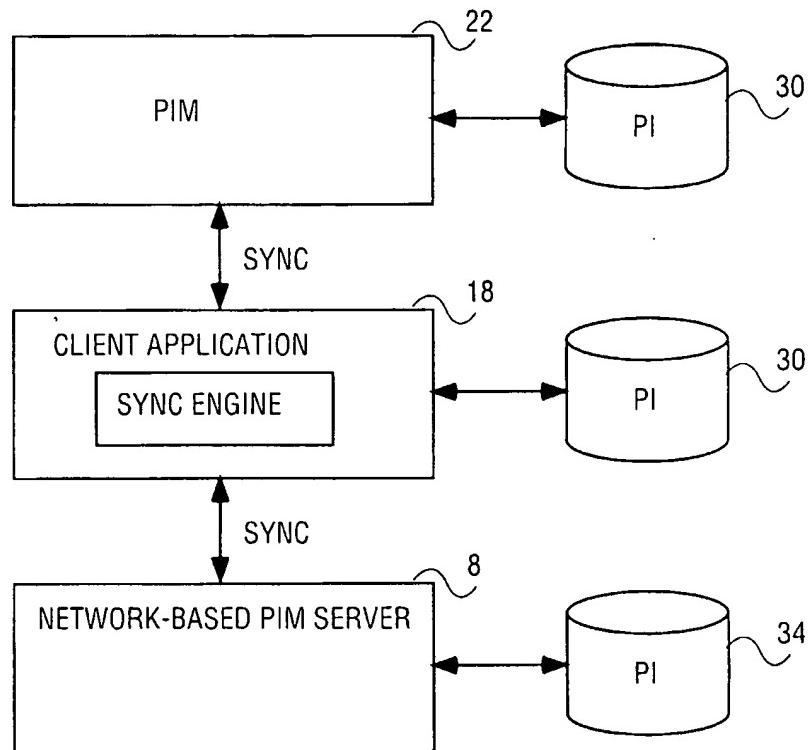


FIG. 9B

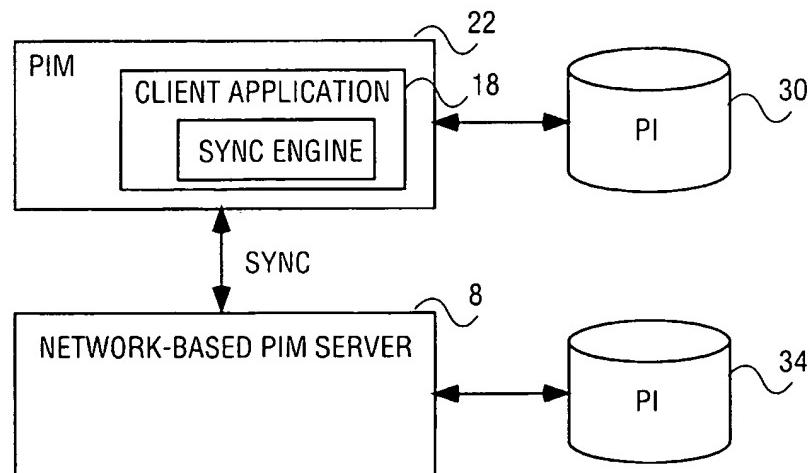


FIG. 9C

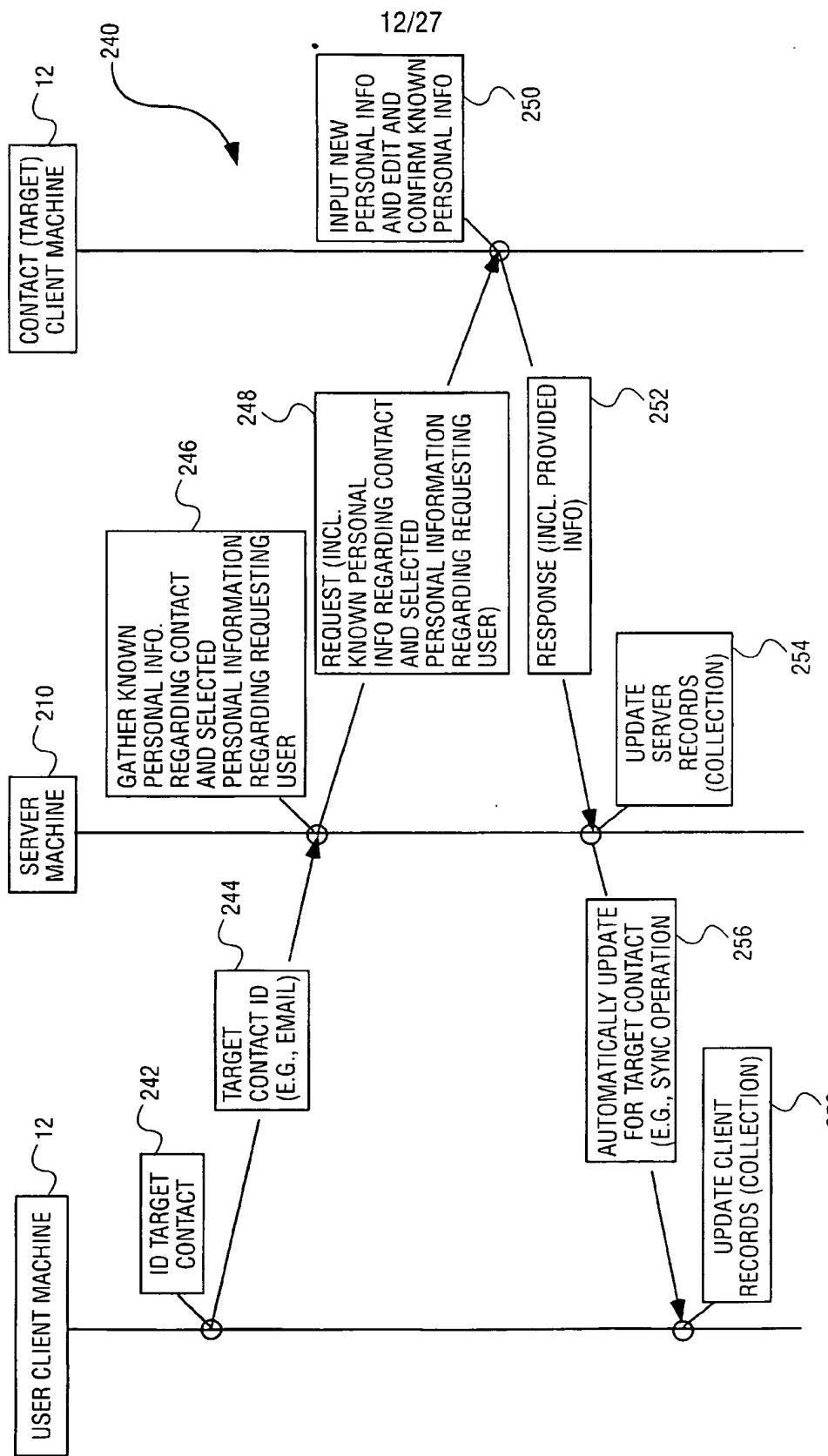
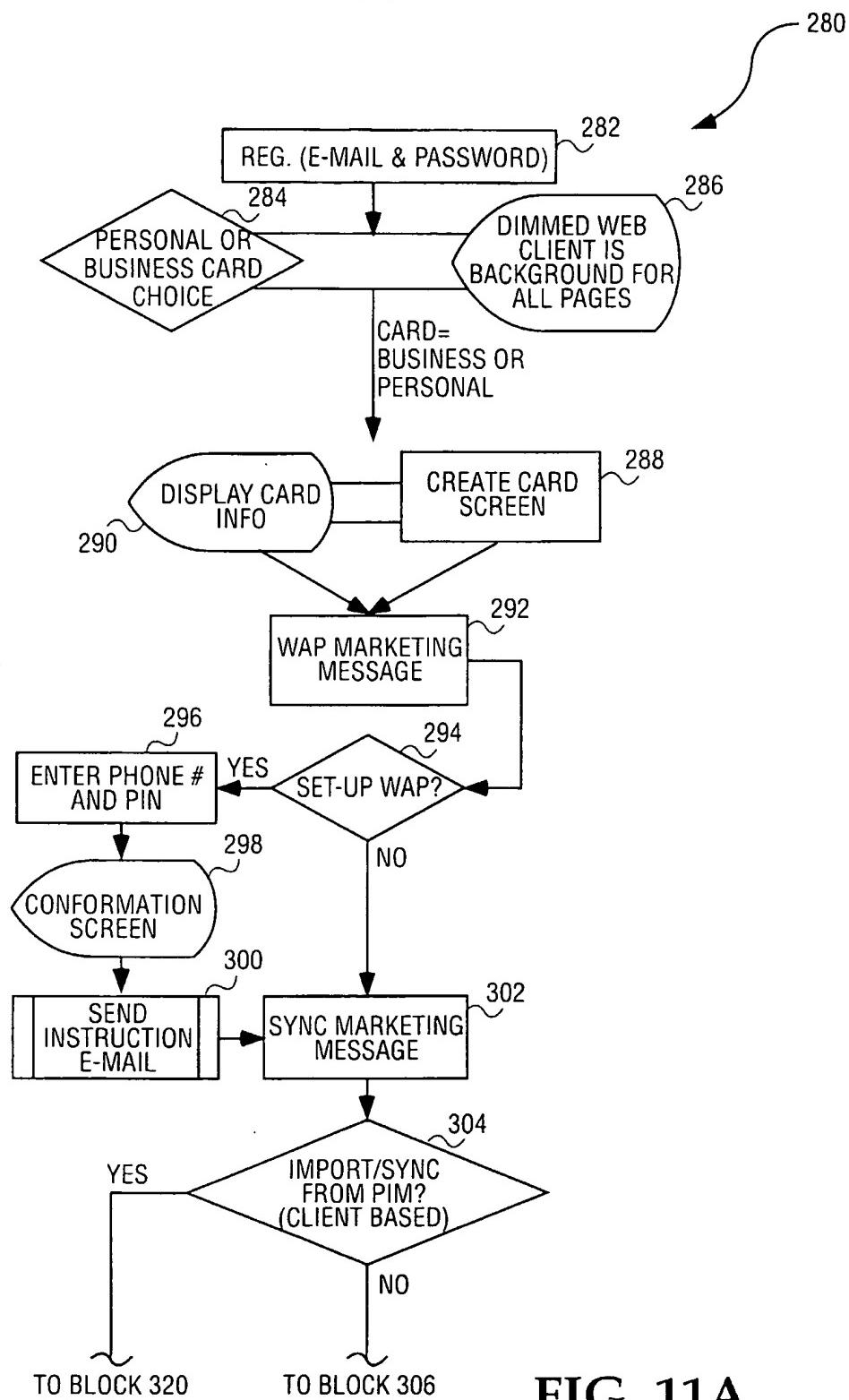


FIG. 10

13/27

**FIG. 11A**

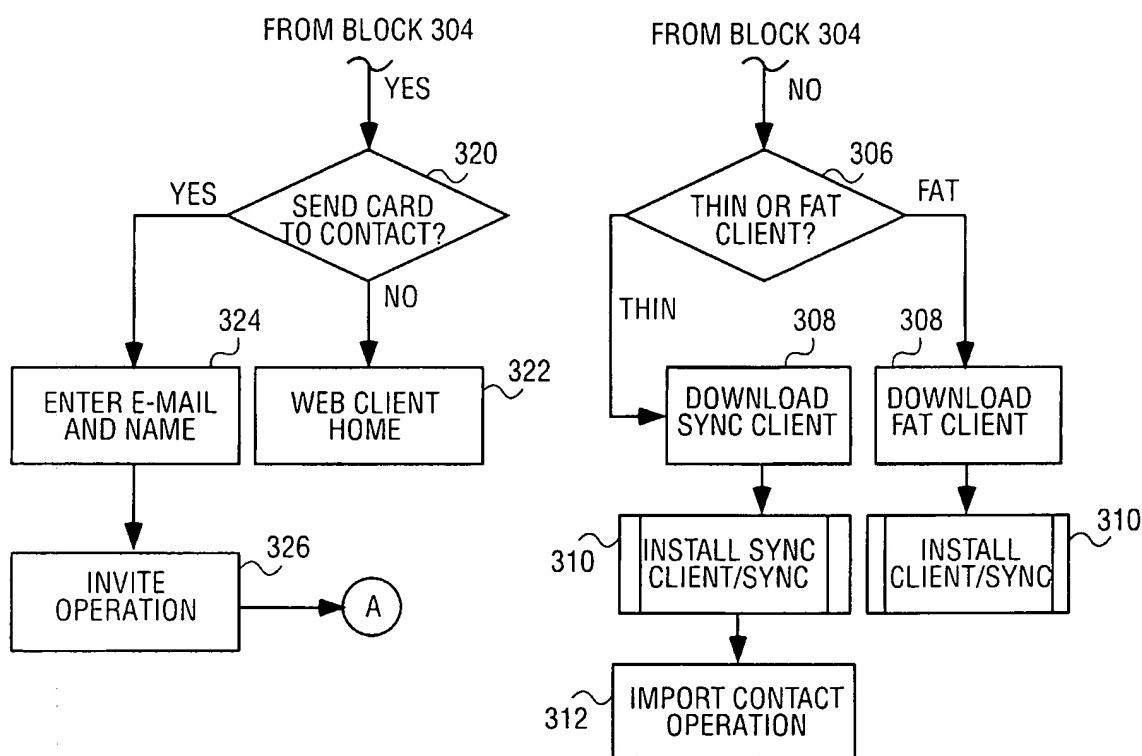
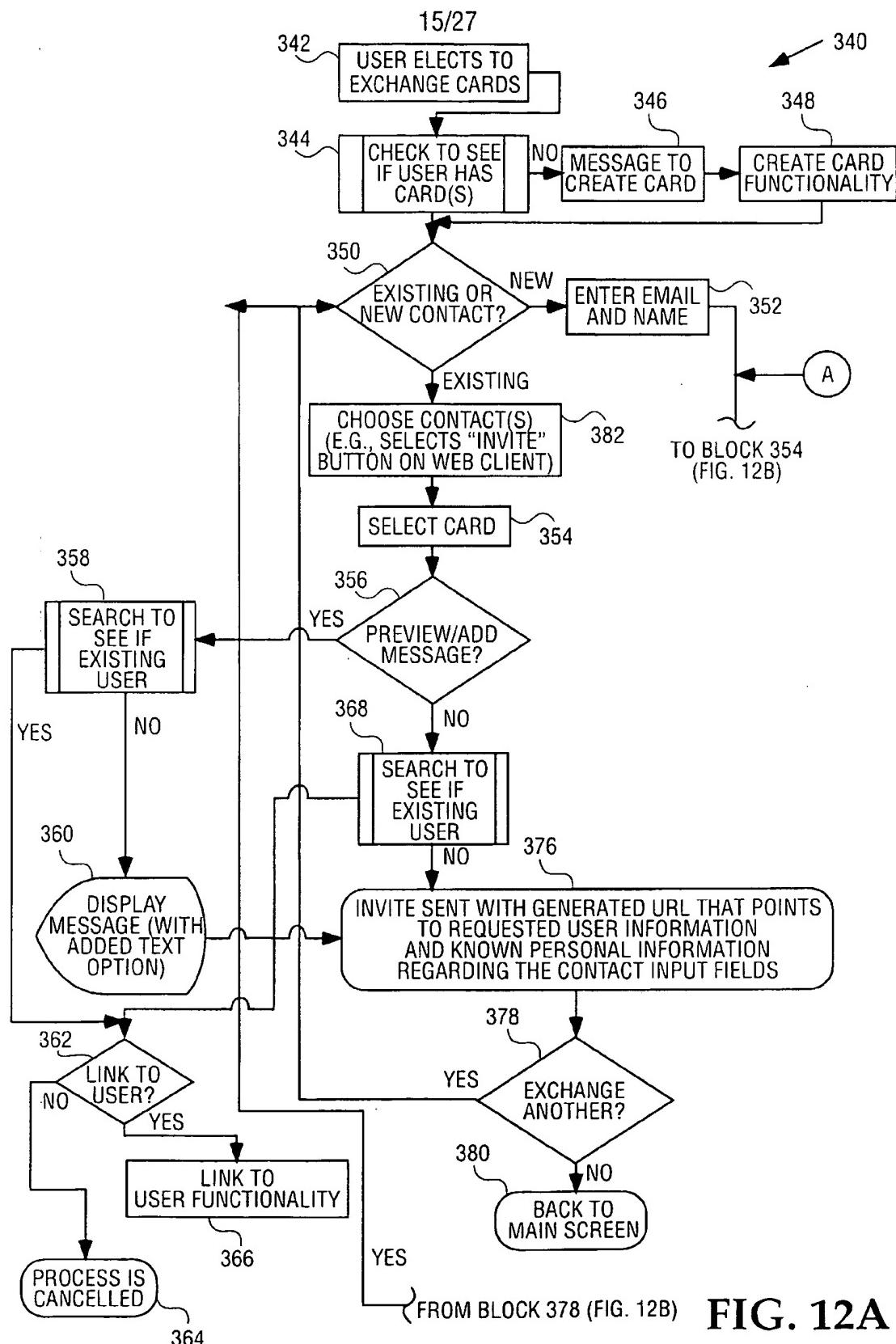
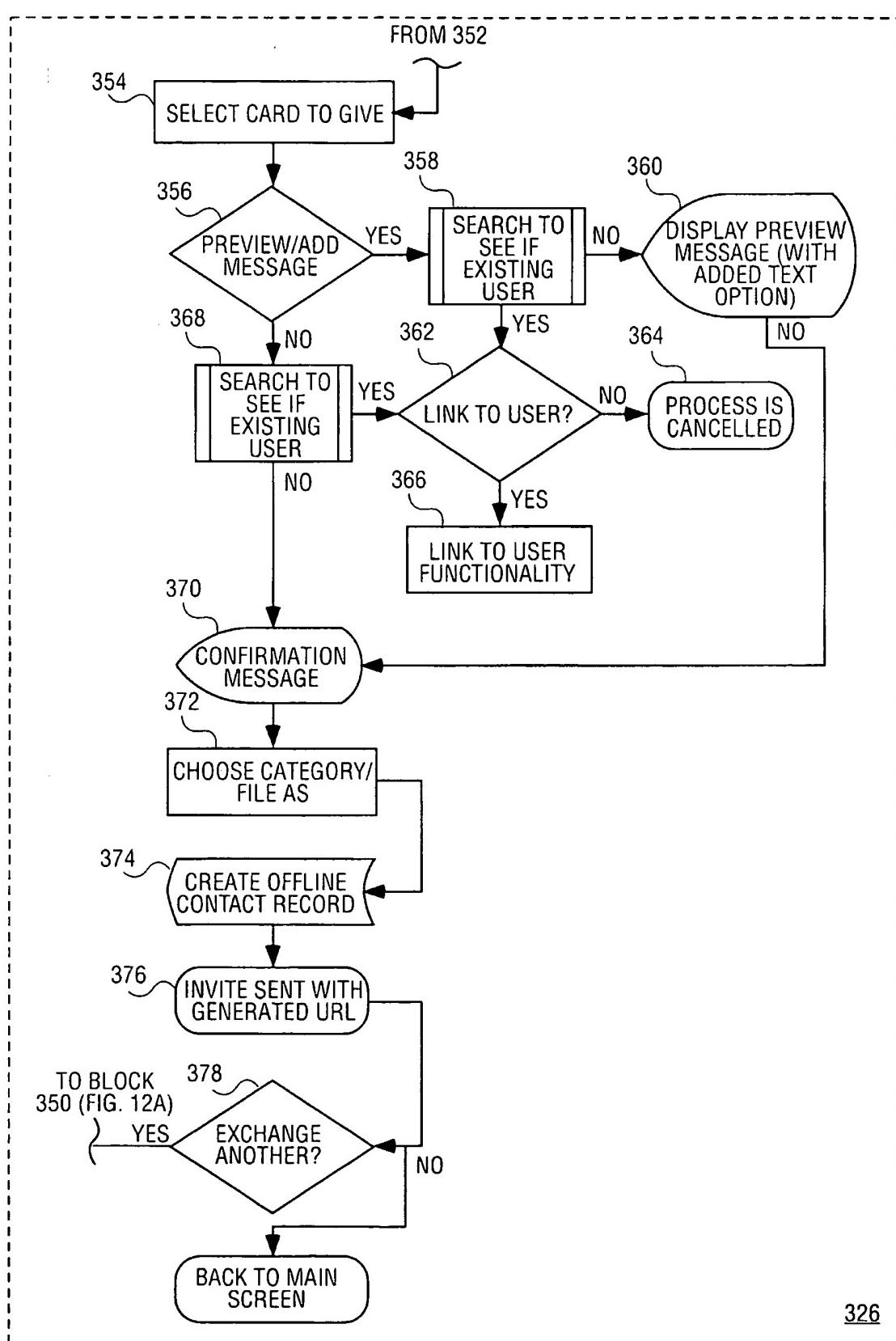


FIG. 11B

**FIG. 12A**

16/27

340



17/27

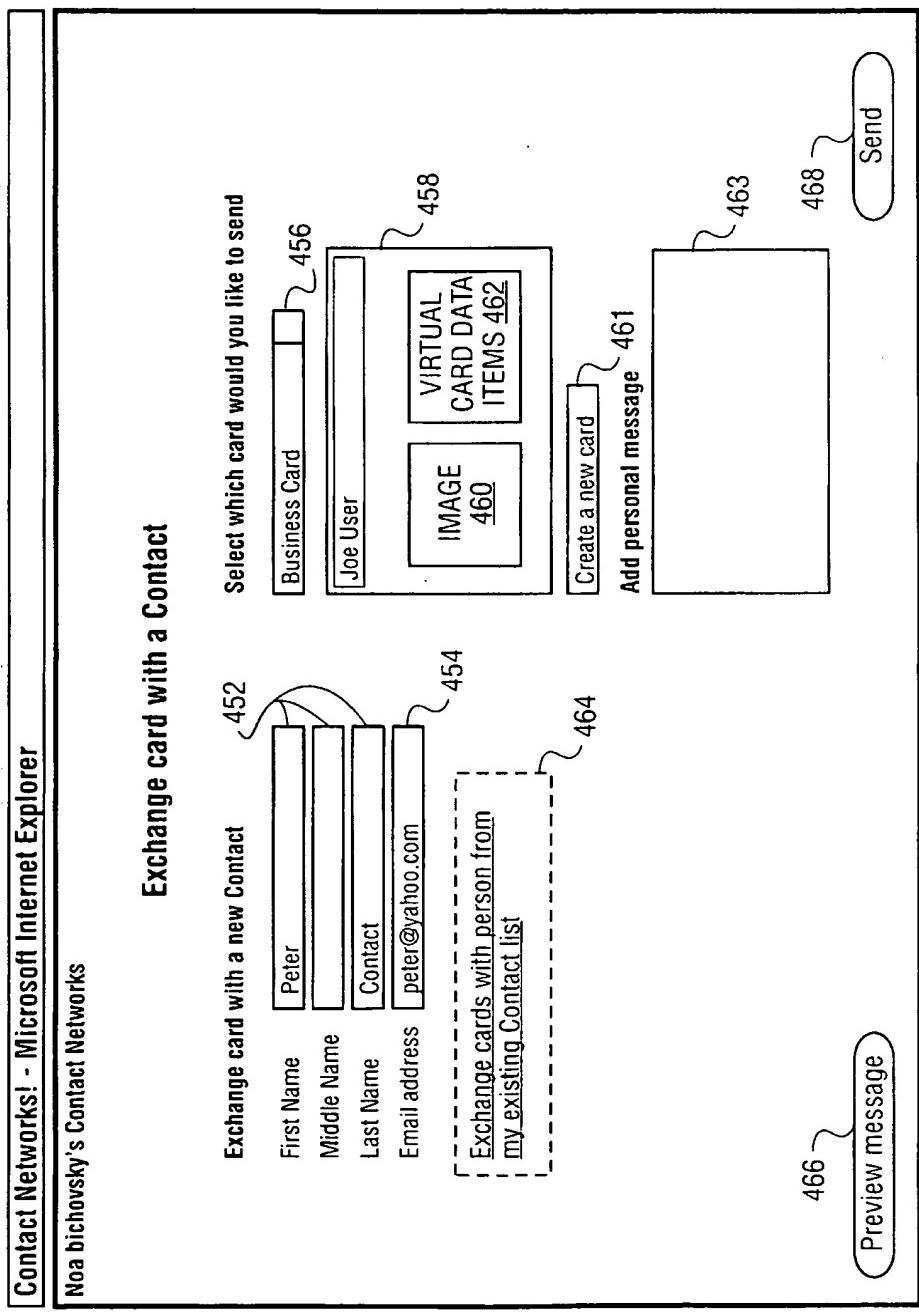


FIG. 13

18/27

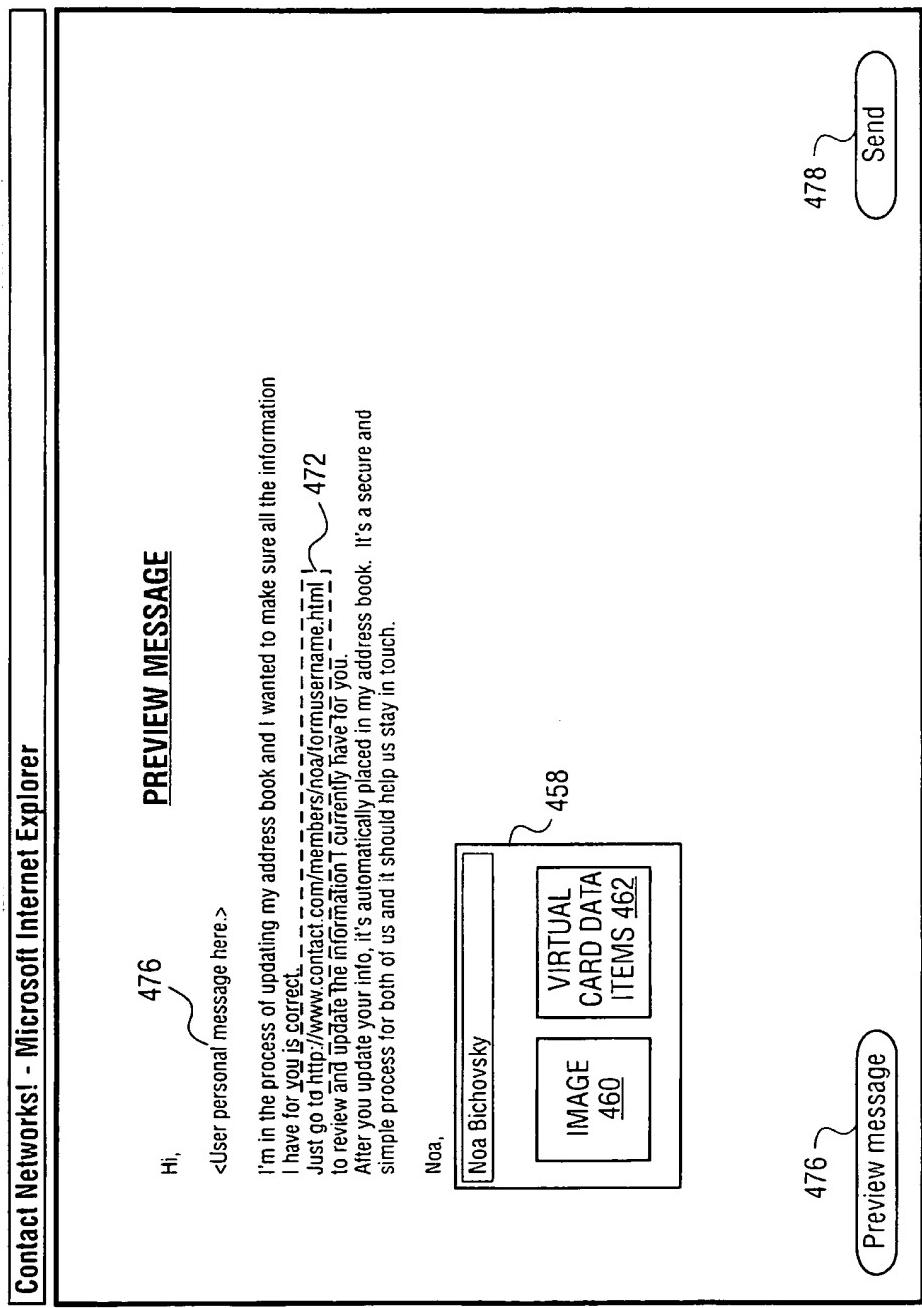


FIG. 14

19/27

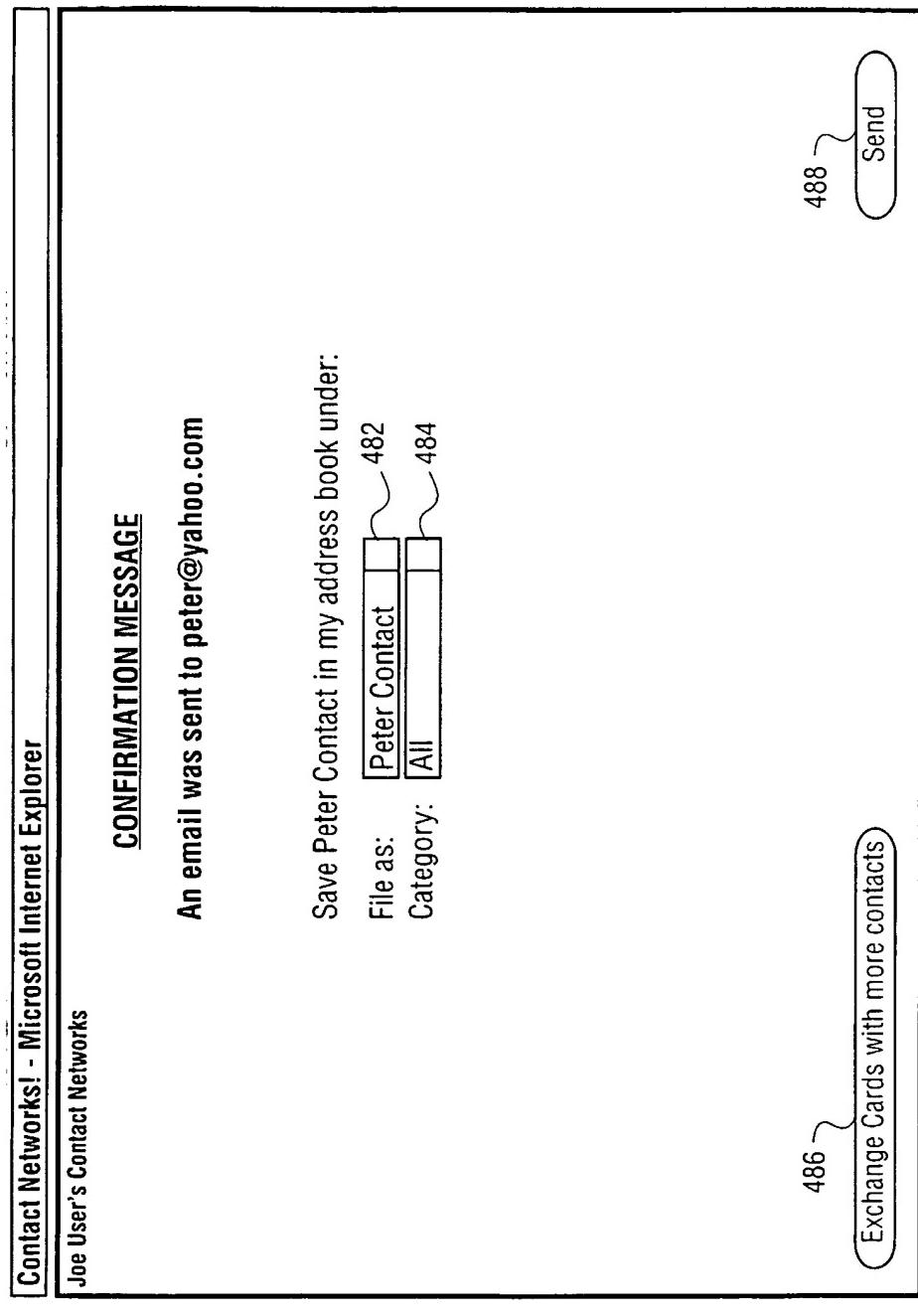


FIG. 15

20/27

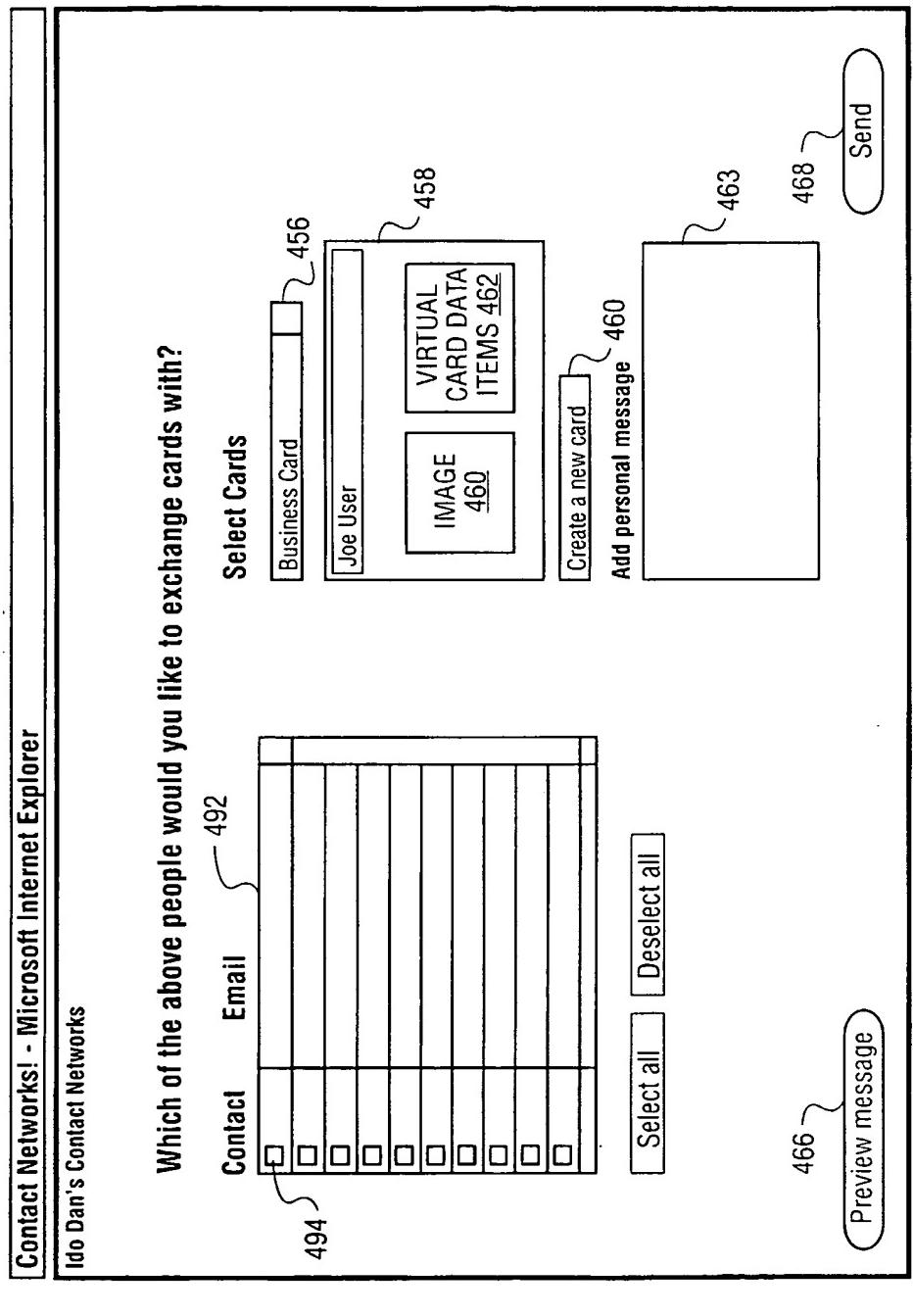


FIG. 16

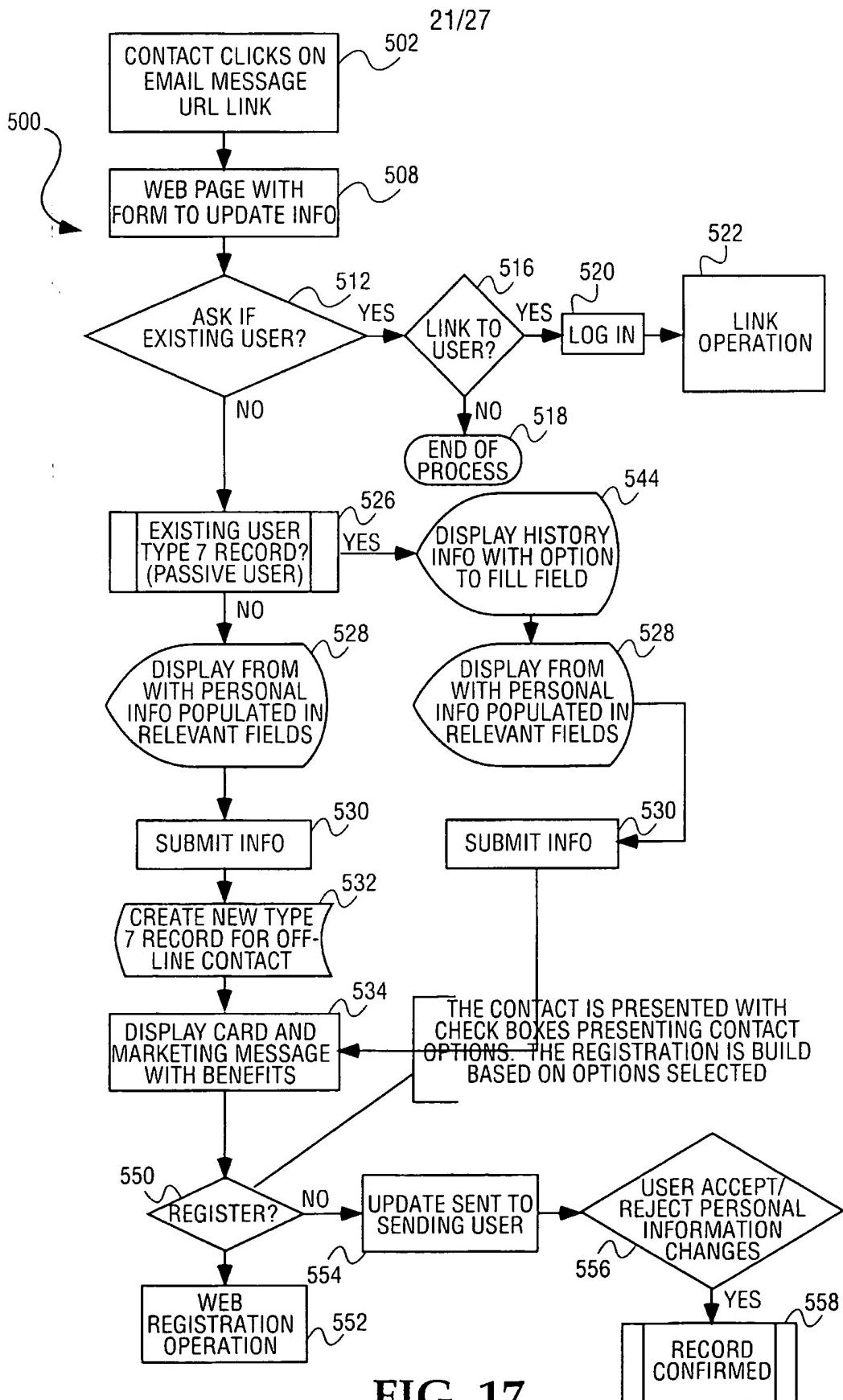
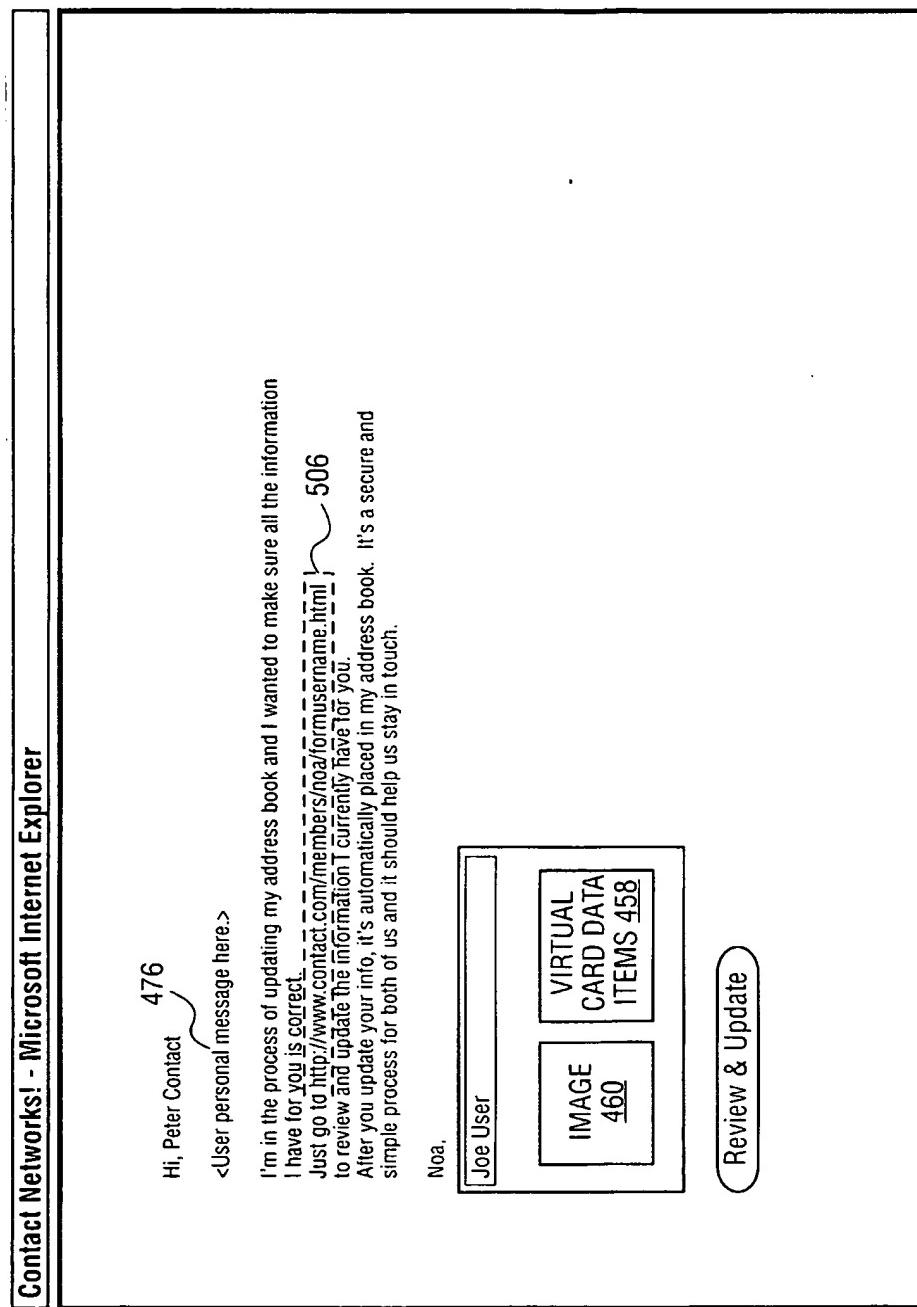


FIG. 17

SUBSTITUTE SHEET (RULE 26)

22/27



504

FIG. 18

23/27

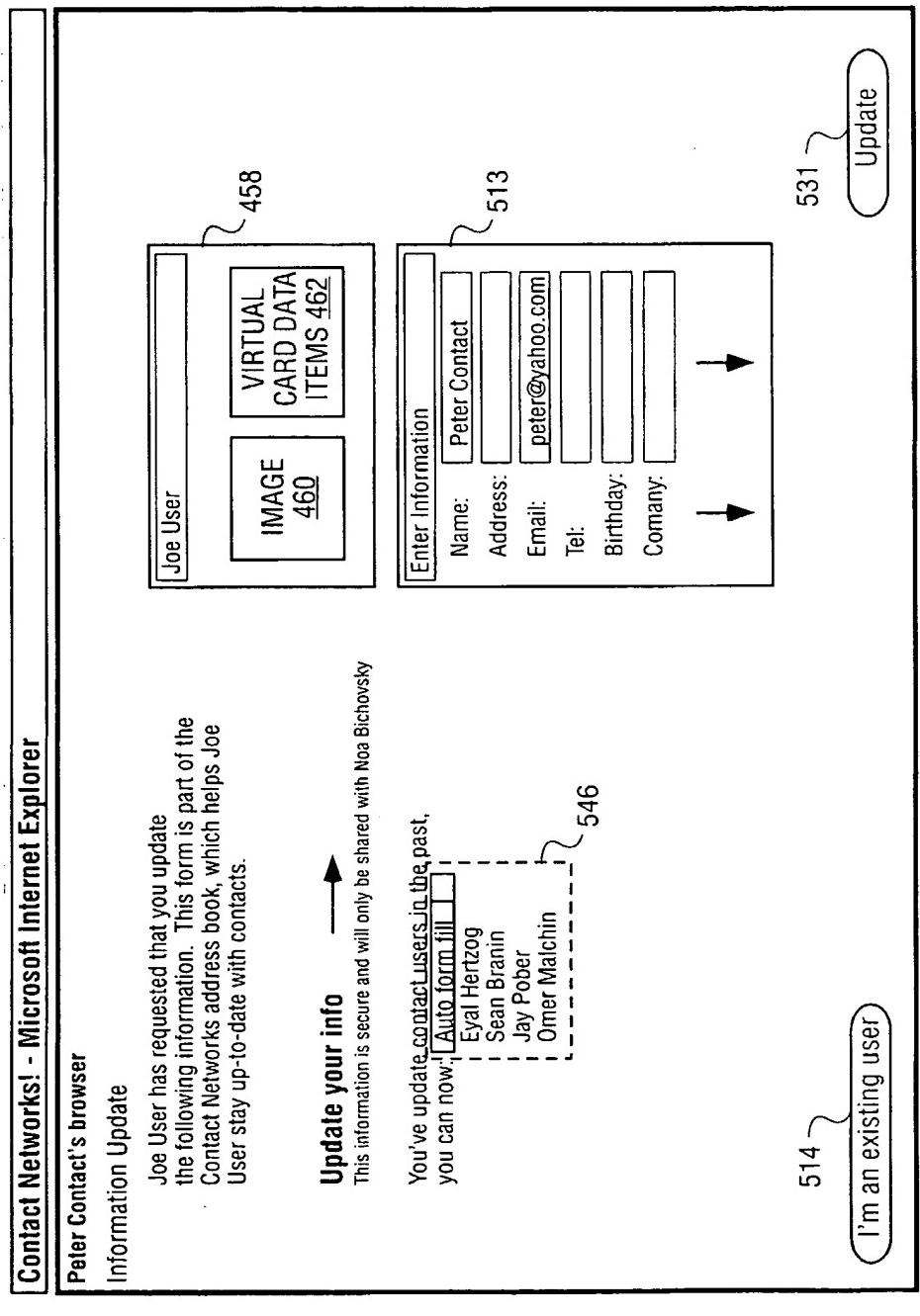
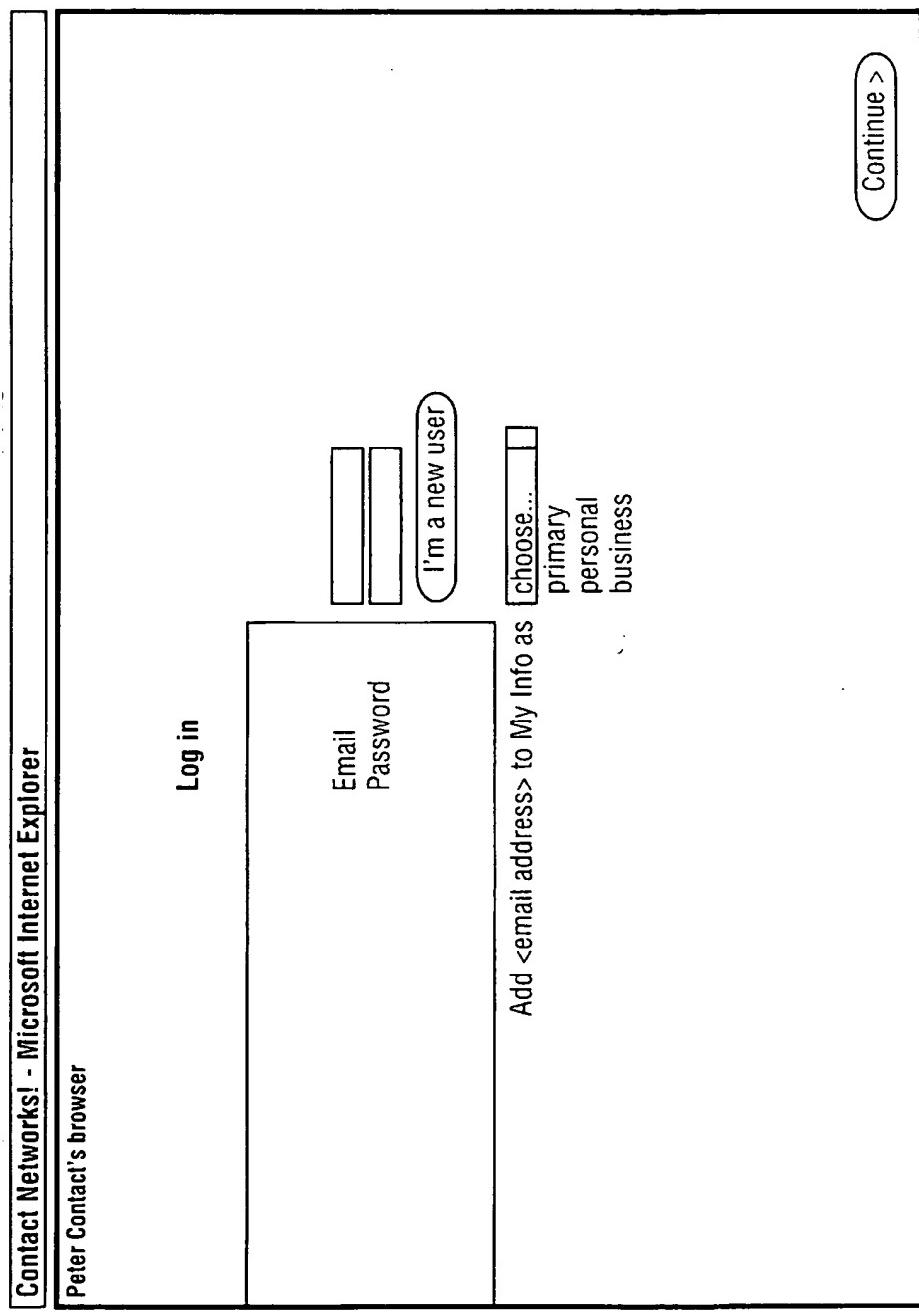


FIG. 19

24/27



524

FIG. 20

25/27

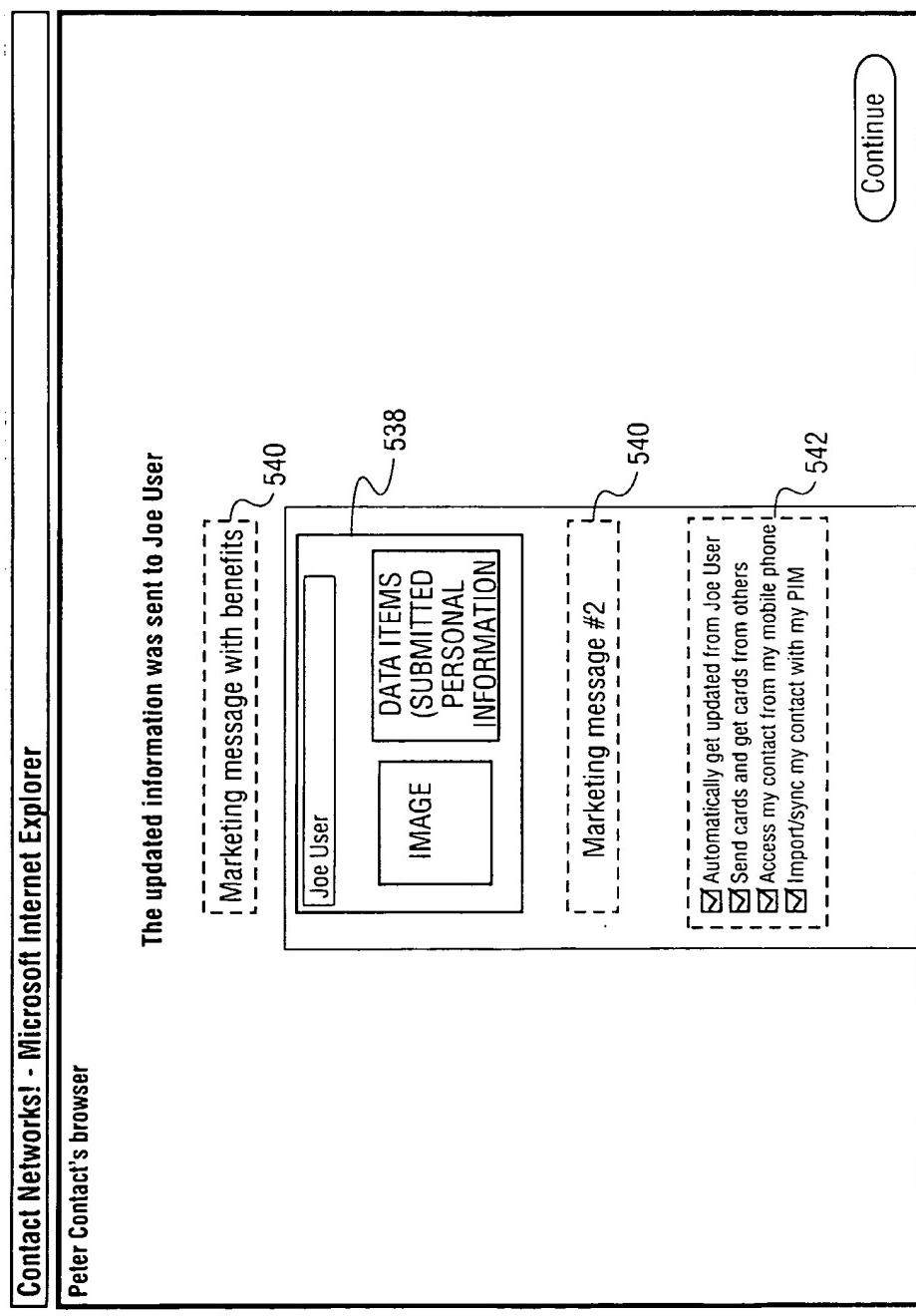


FIG. 21

536

26/27

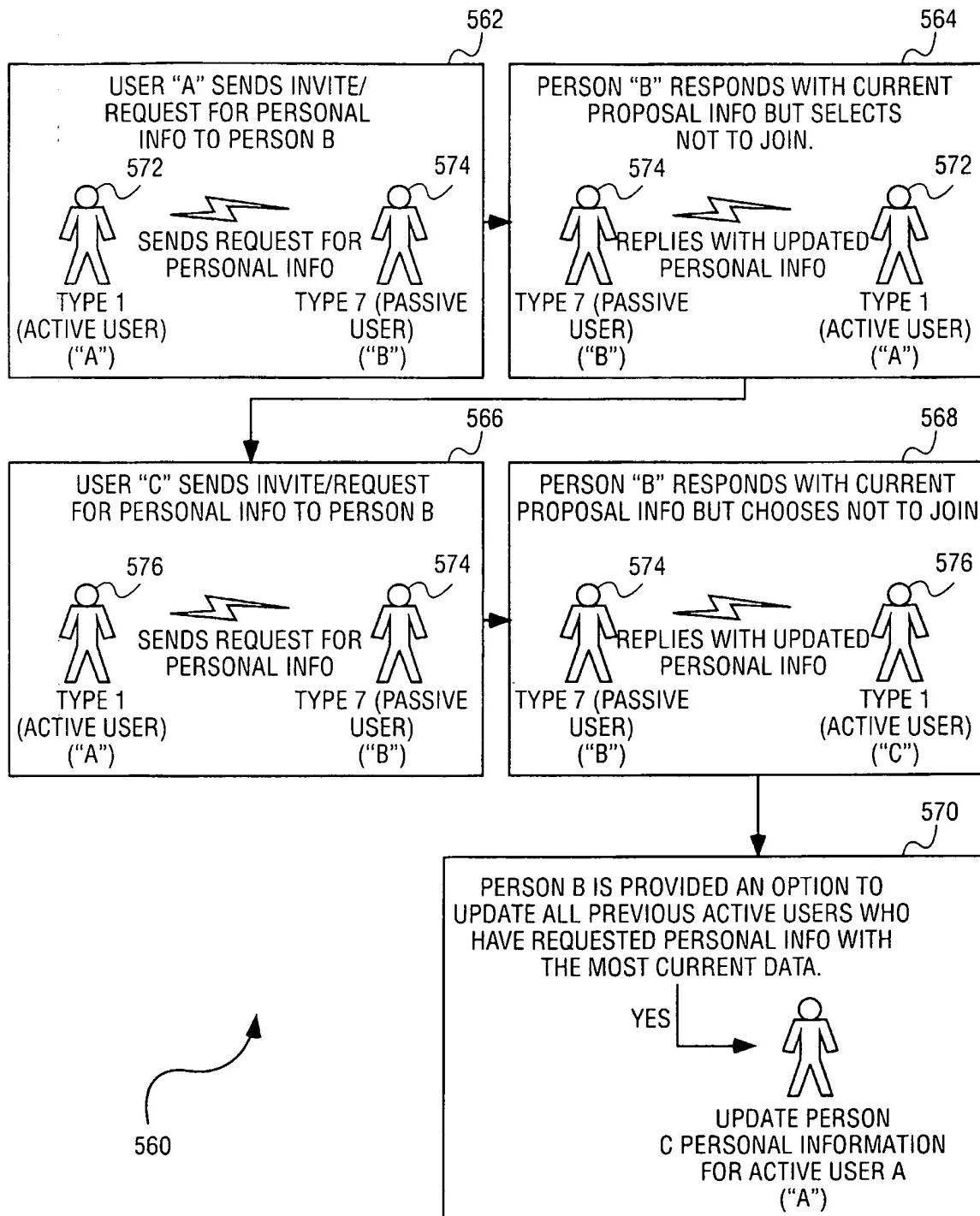
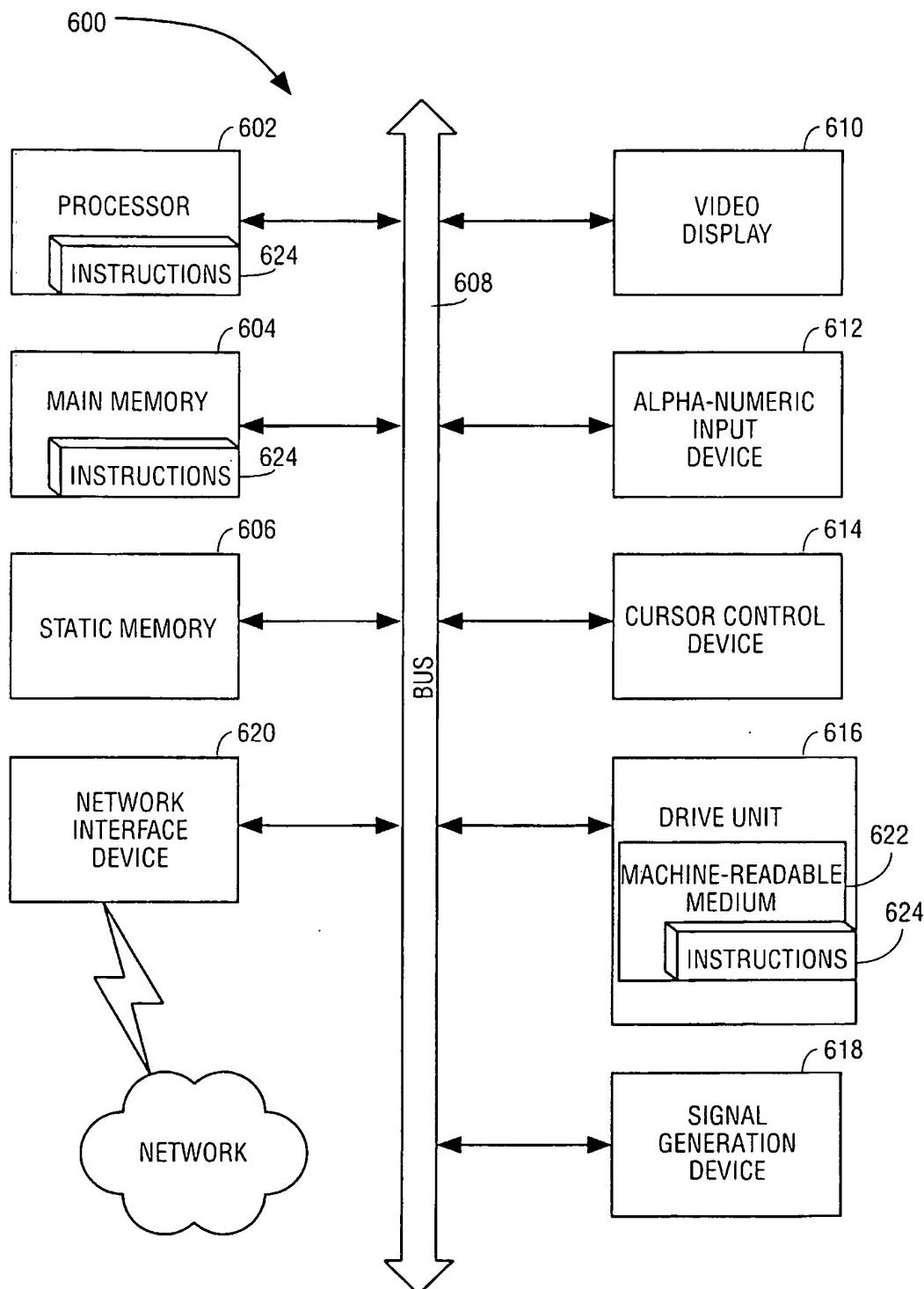


FIG. 22

27/27

**FIG. 23**

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US00/29719

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 17/30, 7/00, 13/00, 3/00

US CL :Please See Extra Sheet.

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/9; 707/10; 345/327

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,813,006 A (POLNEROW et al) 22 September 1998, whole document.	1, 24-25, 37 -----
Y		2-23, 26-36
Y	US 5,950,200 A (SUDAI et al) 07 September 1999, whole document.	2-23, 26-36
A	US 5,945,988 A (WILLIAMS et al) 31 August 1999, whole document.	1-37

Further documents are listed in the continuation of Box C.  See patent family annex.

* Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
*A* document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
*E* earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
*O* document referring to an oral disclosure, use, exhibition or other means		
*P* document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

21 NOVEMBER 2000

Date of mailing of the international search report

09 JAN 2001

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

HEATHER HERNDON

Telephone No. (703) 305-3900

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US00/29719

A. CLASSIFICATION OF SUBJECT MATTER:  
US CL :

707/9; 707/10; 345/327

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**